ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3308

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Equipe Systèmes Intelligents, Réseaux, Génie Logiciel et Algorithmes
Discipline : Informatique
Spécialité : Réseaux

Présentée et soutenue le 20/07/2020 par :

Mounir AZIZI

Gestion des outils OAM et optimisation du routage dans les réseaux MPLS-TP

JURY

Mme. Fouzia OMARY	PES, Faculté des Sciences, Université Med V, Rabat	Présidente
Mr. Farid EL HEBIL	PES, ENSA, Université Mohammed Premier, Oujda	Rapporteur-Examinateur
Mme. Fatima-Zahra BELOUADHA	PES, EMI, Université Mohammed V, Rabat	Rapporteur-Examinateur
Mme. Ghizlane ORHANOU	PH, Faculté des Sciences, Université Med V, Rabat	Rapporteur-Examinateur
Mr. Redouane BENAINI	PH, Faculté des Sciences, Université Med V, Rabat	Directeur de thèse
Mr. Mouad BEN MAMOUN	PES, Faculté des Sciences, Université Med V, Rabat	Co-Directeur

Année Universitaire : 2019/2020

✉ Faculté des Sciences, avenue Ibn Battouta, BP. 1014 RP, Rabat –Maroc
☎ 00212 (05) 37 77 18 76 🖷 00212(05) 37 77 42 61; http://www. fsr.um5.ac.ma

Dédicaces

Au nom du dieu le clément et le miséricordieux louange à ALLAH le tout puissant.

Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement:

A ma mère,

pour m'avoir entouré d'amour, d'invocation et de sacrifice, que dieu te garde,

A la mémoire de mon père,

aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous,

A ma femme,

pour la patience et le soutien dont tu as fait preuve pendant toute la durée de cette thèse,

A mes enfants,

pour l'espoir et l'amour que vous gravez de jour en jour dans mon cœur,

A mes frères et sœurs,

pour votre amour et vos encouragements.

Et à tous ceux qui m'ont soutenu et m'ont conseillé

Remerciements

Les travaux présentés dans le mémoire ont été effectués au sein de l'Équipe Systèmes Intelligents, Réseaux, Génie Logiciel et Algorithmes (ANISSE) à la faculté des sciences de Rabat sous la direction du Professeur Redouane BENAINI et le co-encadrement du Professeur Mouad BEN MAMOUN.

En premier lieu, je tiens à remercier mon Directeur de Thèse le Professeur Redouane BENAINI pour la confiance qu'il m'a accordée, pour ses multiples conseils, pour ses qualités humaines d'écoute et de compréhension et pour toutes les heures qu'il a consacrées à diriger cette recherche. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

J'adresse de chaleureux remerciements au Professeur Mouad BEN MAMOUN pour avoir co-encadré ce travail, pour son attention de tout instant sur mes travaux, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. J'ai pris un grand plaisir à travailler avec lui.

Je tiens à remercier la présidente du jury Madame Fouzia OMARY, Professeure à la Faculté des Sciences à Rabat, qui m'a honoré en acceptant d'être présidente du jury de ma soutenance.

Je tiens également à exprimer ma gratitude Monsieur Farid EL HEBIL, Directeur de l'École nationale des sciences appliquées à Oujda d'avoir accepté d'être rapporteur et examinateur.

Mes gratitudes vont aussi à Madame Fatima-Zahra BELOUADHA, Professeure à l'École Mohammadia d'Ingénieurs à Rabat d'avoir accepté d'être rapporteur et examinateur.

Mes gratitudes vont aussi à Madame Ghizlane ORHANOU, Professeure à la Faculté des Sciences à Rabat d'avoir accepté d'être rapporteur et examinateur.

Je tiens également à exprimer ma gratitude à Monsieur Mohammed BENKHALIFA Responsable de l'Équipe Systèmes Intelligents, Réseaux, Génie Logiciel et Algorithmes de la Faculté des Sciences de Rabat.

Résumé

Les réseaux de transports connaissent depuis plusieurs années un changement capital en passant de technologies traditionnelles à commutation par circuit (TDM) à des réseaux dit de nouvelle génération (NGN) comme Multiprotocol Label Switching Transport Profile (MPLS-TP). Toutefois différents standards d'outils d'Opérations, Administrations et Maintenances (OAM) coexistent ensemble ce qui pose de réel problèmes aux opérateurs de transports réseaux.

Cette thèse présente d'abord les différents outils OAM utilisées dans les réseaux MPLS-TP. Ensuite, elle expose la problématique d'interopérabilité des OAM causée par le fait qu'il y a deux familles de standards différents d'OAM. Après avoir fait l'état de l'art des solutions traitant cette problématique, nous avons proposé deux solutions, intitulées « modèle overlay » et « modèle de cloisonnement », pour résoudre cette problématique Par la suite, nous avons appliqué le paradigme Software Defined Netwrking (SDN) aux réseaux MPLS-TP pour proposer d'exécuter les OAM comme un service réseau. En dernière partie, nous proposons une solution pour utiliser des outils OAM afin d'optimiser les ressources dans des réseaux de transport MPLS-TP basée sur le paradigme SDN. Pour se faire, notre approche se base sur l'utilisation de l'algorithme de colonie de fourmis modifié qui est comparée à d'autres approches existantes. A la fin de ce manuscrit, l'environnement, la méthodologie et les résultats des simulations sont analysées.

Mots-clefs: MPLS-TP, OAM, SDN, Openflow, Algorithme de Colonie de Fourmis.

Abstract

During last years, transport networks have been undergoing a major change by moving from traditional circuit-switched (TDM) technologies to so-called New Generation Networks (NGN) like Multiprotocol Label Switching Transport Profile (MPLS-TP). However, different standards of Operations, Administration and Maintenance (OAM) tools coexist together, which poses real problems for network transport operators.

This thesis first presents the different OAM standard and highlights their interoperability's problem. Then, we have proposed two solutions resolve this problems the first is called "overlay model" and the second is called "partitioning model". We then applied the Software Defined Networking (SDN) paradigm to MPLS-TP networks in order to launch OAM directly as a network service. Finally, we propose a solution to use OAM tools to optimize resources in MPLS-TP transport networks based on the SDN paradigm using a modified ant colony algorithm which is compared to other existing approaches. At the end of this manuscript, the simulation results are extensively detailed and analyzed.

Key Words: MPLS-TP, OAM, SDN, Openflow, Ant-Colony Algorithm

Tables des Matières

DEDICACES **1**

REMERCIEMENTS **2**

RESUME **3**

ABSTRACT **4**

TABLES DES MATIERES **5**

LISTE DES FIGURES **7**

LISTE DES TABLEAUX **9**

LISTE DES ABREVIATIONS **10**

1. INTRODUCTION GENERALE **13**

1.1. Contexte **13**
1.2. Problématique des OAM **14**
1.3. Contributions et publications **15**
1.4. Structure de la thèse **15**

2. LES RESEAUX DE TRANSPORT BASES SUR MPLS-TP **17**

2.1. Introduction **17**
2.2. Introduction aux réseaux de transport **17**
2.2.1. Granularité de la bande passante et flexibilité des débits 20
2.2.2. Qualité de service QoS 20
2.3. Les réseaux de transport Ethernet (Carrier Ethernet) **21**
2.3.1. Définitions et caractéristiques 21
2.3.2. Réseaux de transport de nouvelles génération 23
2.3.3. Généralités sur la technologie MPLS 24
2.3.4. Les principes de la technologie MPLS-TP 28
2.4. Conclusion **34**

3. LES OUTILS D'OPERATION, D'ADMINISTRATION ET DE MAINTENANCE **35**

3.1. Introduction **35**
3.2. Les Outils OAM **35**
3.2.1. Les OAM Ethernet : définis par ITU-T 37
3.2.2. Les OAM MPLS-TP: définis par IETF 44
3.3. Etat de l'art sur les solutions d'interopérabilité OAM existantes **47**
3.3.1. La problématique des outils OAM dans les réseaux MPLS-TP 47
3.3.2. Solutions existantes pour l'interfonctionnement entre les différents outils OAM 48
3.3.3. Utilisation du plan de Management par le plan de contrôle 51
3.4. Conclusion **52**

4. PROPOSITION DE MODELES POUR RESOUDRE LES PROBLEMATIQUES D'INTEROPERABILITE OAM DANS LES RESEAUX MPLS-TP 53

4.1. Introduction 53
4.2. Proposition de Modèle Overlay 53
4.2.1. Le concept du modèle Overlay 54
4.2.2. L'application du modèle Overlay aux réseaux de transport MPLS-TP 55
4.3. Proposition de Modèle de Cloisonnement 56
4.3.1. Concept du modèle de Cloisonnement statique 57
4.3.2. Modèle de cloisonnement « nœud de frontière » : 58
4.3.3. Modèle de cloisonnement « Segment de frontière » : 59
4.3.4. Variante du modèle de cloisonnement « Segment de frontière » : ODM 59
4.3.5. Evaluation des solutions proposées : simulations et résultats 62
4.4. Conclusion 67

5. UTILISATION DU CONCEPT SDN POUR LA GESTION DES OAM DANS LES RESEAUX MPLS-TP 69

5.1. Présentation du paradigme SDN 69
5.1.1. Introduction 69
5.1.2. Architecture SDN 70
5.2. Adoption du Paradigme SDN dans les réseaux MPLS-TP 74
5.3. Implémentation d'une solution de mesure de délai basée sur SDN 76
5.3.1. Fonctionnement du module « Mesure de délai » RTT sur un modèle SDN 76
5.3.2. Concept du module OAM proposé 77
5.3.3. Evaluation de la solution proposée 80
5.4. Conclusion 84

6. PROPOSITION DE SOLUTION D'OPTIMISATION DE ROUTAGE BASEE SUR SDN ET SUR LES OUTILS OAM 85

6.1. Introduction 85
6.2. Optimisation du routage dans les réseaux de transports 85
6.2.1. Algorithme Dijkstra 86
6.2.2. Algorithme de Colonie de fourmis 88
6.2.3. Modèle amélioré de l'algorithme de Colonie de fourmis : ACO-OAM 89
6.3. Tests et Résultats des simulations 93
6.3.1. Environnement des expérimentations 93
6.3.2. Méthodologie des tests et des critères d'évaluations 95
6.3.3. Analyse des résultats et conclusions 96
6.4. Conclusion 99

7. CONCLUSION GENERALE 100

BIBLIOGRAPHIE 102

ANNEXE A : MININET 107

ANNEXE B : CONTROLEUR RYU 111

Liste des figures

Figure 1 Trafic IP mondial entre 2013 et 2018 18
Figure 2 Les tendances des revenus des opérateurs par type de service 18
Figure 3 Passage du mode circuit au mode paquet avec la technologie Ethernet dans les réseaux mobiles 19
Figure 4 Évolution de la transition des équipements TDM vers l'Ethernet 19
Figure 5 Evolution des technologies selon les organismes de standardisation 23
Figure 6 PBB-TE : Exemple de transfert de trame Ethernet 23
Figure 7 LER/LSR : interaction entre les différentes composantes MPLS 25
Figure 8 MPLS-TP: un mélange de sous-ensemble de MPLS et de nouvelles fonctionnalités orientées transport 28
Figure 9 Architecture MPLS-TP: multiple Pseudowire 29
Figure 10 Architecture MPLS-TP: multiple tunnels transportés 29
Figure 11 MPLS-TP : Les attributs majeurs 30
Figure 12 Plan de données MPLS-TP : procédures supportées 31
Figure 13 Le plan de contrôle dynamique du MPLS-TP 32
Figure 14 MPLS-TP : Mécanismes de protection et de restauration 33
Figure 15 Format d'un paquet OAM 36
Figure 16 Les OAM Ethernet 38
Figure 17 CFM : Concept des Points de Maintenance 38
Figure 18 CFM : Multi-Domain vs Association de Maintenance MA 39
Figure 19 supervision d'un VLAN à l'aide de CFM 40
Figure 20 Messages Loopback 41
Figure 21 Messages Linktrace 41
Figure 22 Positionnement des OAM EFM au niveau de la couche 2 du modèle OSI 43
Figure 23 Les OAM MPLS-TP définis par IETF 44
Figure 24 Ping LSP 45
Figure 25 Traceroute LSP 45
Figure 26 BFD : Machine d'état 46
Figure 27 Format des trames PDU OAM: IETF vs ITU-T 47
Figure 28 Illustration de la couche « fine » à travers plusieurs réseaux de Transport 49
Figure 29 Illustration du rajout de la couche « fine » sur les couches d'un réseau de Transport .. 49
Figure 30 Schéma de principe d'un nœud IW 50
Figure 31 Architecture PCE 51
Figure 32 Principe d'un réseau overlay 54
Figure 33 Modèle Overlay proposé 54
Figure 34 PW de transport : base du modèle Overlay proposé 55
Figure 35 Modèle Overlay proposé : superposition des couches grâce aux PW 55
Figure 36 Différents niveaux de maintenance du modèle Overlay proposé 56
Figure 37 Cloisonnement en sous réseaux 57
Figure 38 Principe MS-PW et « LSP Stitching » 58
Figure 39 Modèle de cloisonnement proposé: « nœud de frontière » 58

Figure 40 Modèle de cloisonnement proposé: « Segment de frontière » 59
Figure 41 Processus d'établissement du mécanisme ODM proposé..... 61
Figure 42 Déroulement du mécanisme ODM proposé 61
Figure 43 Schéma de la maquette du test de fonctionnement du modèle proposé 62
Figure 44 Configuration des éléments de la maquette 63
Figure 45 Construction des MEP et MIP dans le modèle de cloisonnement statique 64
Figure 46 Exemple de trace VCCV au niveau du pseudowire..... 64
Figure 47 Propagation d'OAM inter-segment d'un service E-line construit en MS-PW 66
Figure 48 Schéma de la maquette: « Segment de frontière » 66
Figure 49 Vue sous forme de couches du paradigme SDN 70
Figure 50 Vue globale sur l'architecture SDN..... 71
Figure 51 Parcours d'un paquet au niveau du pipeline..... 73
Figure 52 Les équipements de transfert SDN fonctionnant avec OpenFlow 74
Figure 53 Utilisation du PCE pour contrôler un réseau MPLS-TP 75
Figure 54 Module OAM pour le calcul de délai de transit sur un chemin..... 77
Figure 55 L'ensemble des délais impliqués dans le calcul du délai de transit aller-retour..... 78
Figure 56 règles et actions de chaque table de flux du module OAM 80
Figure 57 Maquette de simulations..... 80
Figure 58 iperf serveur et client 82
Figure 59 Topologie de référence pour les simulations..... 82
Figure 60 Test #1 : une instance OAM par LSP..... 83
Figure 61 Test #2 : une instance OAM par LSP (nombre de LSP: 2, nombre de hops: 3, 5 et 10) 83
Figure 62 Test #3: une instance OAM par LSP (nombre de LSP : 5, nombre de hops: 10) 84
Figure 63 Exemple du déroulement de l'algorithme Dijkstra..... 87
Figure 64 Comportement des fourmis pour rejoindre leur nourriture 88
Figure 65 Charte de l'algorithme ACO-OAM..... 92
Figure 66 Architecture logique de notre banc d'essai..... 93
Figure 67 Topologies utilisées pendant les différents essais 94
Figure 68 Temps de convergence par algorithme..... 96
Figure 69 Nombre de sauts par chemin 97
Figure 70 Délai de bout en bout sur un LSP en présence de contraintes SLA: débit à 10 Mbps.. 97
Figure 71 Délai de bout en bout sur un LSP en présence de contraintes SLA: débit à 40 Mbps.. 98
Figure 72 Délai de bout en bout sur un LSP en présence de contraintes SLA: débit à 70 Mbps.. 98
Figure 73 Taux de succès de chaque algorithme en cas de présence de SLA 98

Liste des tableaux

Tableau 1 Les encapsulations standards des services émulés dans MPLS 26
Tableau 2 Attributs des ressources d'un LSP 27
Tableau 3 Les deux grandes classes d'OAM (IETF) 37
Tableau 4 Terminologie des normes Y.1731 et 802.1ag 42
Tableau 5 Composantes d'une table de flux 72
Tableau 6 caractéristiques SDN et MPLS-TP 75
Tableau 7 Les topologies utilisées pendant les simulations 95

Liste des abréviations

API	Application Programming Interface
ATM	Asynchrones Transfer Mode
BFD	Bidirectional Forwarding Detection
BGP	Border Gateway Protocol
CFM	Connectivity Fault Management
Dpid	Datapath id
DS	Differentiated Services
DSL	Domain Specific Language
ECN	Explicit Congestion Notification
ESP	Encrypted Security Payload
FE	Forwarding Element
FL	Floodlight
ForCES	Forwarding and Control Element Separation
FW	Firewall
GRE	Generic Routing Encapsulation
GSMP	General Switch Management Protocol
ICMP	Internet Control Message Protocol
ICO	Inter-controller Communication Overhead
IDC	International Data Corporation
IETF	Internet Engineering Task Force
ITU-T	International Telecommunication Union-Telecommunication
ITW	Interworking
LAN	Local Area Network
LB	Load Balancing
LFB	Logical Function Blocks
LISP	Locator/Identifier Separation Protocol
LLDP	Link Layer Discovery Protocol
LMI	Local Management Interface
LSP	Label Switched Path
LTE	Long Term Evolution

MEP	Maintenance Endpoint Point
MIP	Maintenance Intermediate Point
MPLS	Multiprotocol Label Switching
MPLS-TP	Multiprotocol Label Switching Transport Profile
MS-PW	Multi Segment Pseudowire
NGN	Next Generation Networks
OAM	Operation, Administration and Management
ODL	OpenDayLight
ODM	OAM Discovery Mechanism
OnePK	Open Network Environment Platform Kit
ONF	Open Network Fundation
ONOS	Open Network Operating System
OSGI	Open Services Gateway initiative
OSPF	Open Shortest Path First
OVS	Open VSwitch
OXM	OpenFlow eXtensible Match
PE	Provider Edge
PBB	Provider Backbone Bridges
PBB-TE	Provider Backbone Bridges Traffic Engineering
PCE	Path Computation Element
PCEP	Path Computation Element Communication Protocol
PN	Programmable Network
PW	Pseudowire
PWE3	Pseudowire Emulation Edge to Edge
QoS	Quality of Service
Rest	Representational State Transfer
RFC	Request for Comments
RTT	Round Trip Time
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
SLA	Service Level Agreement
SDC	Software Defined Computing
SDH	Synchronous Digital Hierarchy
SDN	Software Defined Networks

SONET	Synchronous Optical Network
SPOF	Single Point Of Failure
T-LDP	Targeted Label Distribution Protocol
TDM	Time-Division Multiplexing
TLV	Type Length Value
VCCV	Virtual Circuit Connectivity Verification
VLAN	Virtual Local Area Network
VM	Virtual Machine
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

Chapitre 1

Introduction générale

1.1. Contexte

L'expansion fulgurante que connait le monde des technologies d'information incite les opérateurs à avoir des réseaux flexibles et fiables permettant de commercialiser et de lancer des services innovants et profitables dans des courts délais. Les services comme IPTV (Internet Protocol TV), téléphonie sur IP (Internet Protocol), et télé-enseignement ont des prérequis très stricts en termes de bande passante, de délai, de perte de paquets, et de résilience. Ces services sont basés sur le protocole IP au niveau de leur couche paquet et se basent aussi sur la fiabilité de la couche transport pour garantir une meilleure Qualité d'Expérience (QoE) [1].

Il est donc primordial que la technologie de transport soit en mesure de garantir une meilleure QoE tout en étant flexible et profitable. Plusieurs technologies existent actuellement pour assurer cette fonction de transport. Les deux familles les plus répondues sont : les technologies de type commutation par circuit et les autres de type commutation par paquet [2].

Les technologies comme Frame Relay et SDH/SONET (Synchronous Digital Hierarchy/ Synchronous Optical Network) sont des exemples de technologie de transport à commutation par circuit ; elles sont, certes, des technologies matures et offrant des délais de convergence très performants en dessous de 50 ms, mais elles comportent aussi certaines limitations. C'est surtout lié à leur manque de flexibilité et d'évolutivité pour satisfaire la demande en perpétuelle croissance en termes de bande passante, et aussi lié au coût d'investissement associé [3].

Par ailleurs, la simplicité de la technologie Ethernet, son coût très faible et sa conception adaptée au transport de paquets, à la fois dans les réseaux LAN et WAN, font d'elle un candidat idéal pour être la technologie de transport des réseaux de nouvelle génération. En effet, la technologie Ethernet dispose de plusieurs avantages en comparaison avec les technologies SONET/SDH, on pourra citer:

- Efficacité d'utilisation de la bande passante,
- Granularité plus riche en termes de débit,
- Protection de la bande passante avec des mécanismes de Qualité de Services QoS comme le Shaping,

- Topologie très flexible à l'aide de protocole comme Spanning Tree,
- Gestion et maintenance facile,
- Bas coûts

Les Transporteurs Ethernet, appelés aussi « Carrier Ethernet » ou opérateurs, ont retenu deux technologies de transport Ethernet:

- PBB-TE : Provider Backbone Bridge – Traffic Engineering: qui découle du standard IEEE 802.1ad (nommé aussi Mac-In-Mac) avec plus de fonctionnalités, comme l'ingénierie de trafic , facilitant l'approvisionnement par les opérateurs [4].
- MPLS-TP: Multiprotocol Label Switching Transport Profile qui réutilise quelques fonctionnalités MPLS, mais aussi comprend les mécanismes OAM tant désirés par les opérateurs [5].

A ce jour, le PBB-TE a été standardisé sous IEEE 802.1Qay mais n'est utilisé que dans un cadre très restreint en conjonction avec les réseaux IP/MPLS. Par contre les transporteurs Ethernet ont plus recours à MPLS-TP qui présente plusieurs avantages offrant ainsi une vraie solution pour les réseaux de Transport Ethernet.

Cette thèse se focalise sur MPLS-TP comme technologie de Transport et traite les défis liés à un composant essentiel de la technologie qui est les outils d'Opérations, d'Administrations et de Maintenances OAM.

1.2. Problématique des OAM

On ne peut parler d'un réel candidat pour remplacer les anciennes technologies réseaux sans qu'il soit au moins capable de délivrer le même niveau de performance et de fournir au moins les mêmes fonctionnalités de maintenance et de gestion.

Parmi les technologies de transport les plus adaptés aux réseaux de transport de nouvelle génération on retrouve MPLS Transport Profile MPLS-TP. Cette technologie est la résultante du meilleur des deux mondes : celui du MPLS et celui des technologies de transport conventionnelles comme SDH et SONET. Cette technologie est conçue dans le but de profiter des avantages des réseaux de transport à commutation de paquet tout en délivrant le même niveau de disponibilité et de richesse OAM que ceux des réseaux de transport classiques à commutation de circuit. C'est pour cette raison, que MPLS-TP dispose d'un outillage OAM très développé.

Toutefois, cette richesse d'OAM a conduit les deux organismes leaders dans la standardisation de protocoles réseaux, IETF (Internet Engineering Task Force) et ITU-T (International Telecommunication Union-Telecommunication), à imposer chacun son standard. Par conséquent, deux standards OAM MPLS-TP ont vu le jour.

L'UIT-T s'appuie sur la norme Y.1731 pour définir les standards OAM de la technologie MPLS-TP, tandis que l'IETF a étendu sa famille, déjà existante, d'OAM MPLS et qui est basée essentiellement sur le protocole BFD (Bidirectional Forwarding Detection).

Cette situation est problématique pour les opérateurs ou équipementiers qui désirent adopter MPLS-TP comme technologie de transport, puisqu'ils sont confrontés à la fois aux problématiques d'Interfonctionnement entre MPLS-TP et les technologies existantes

comme MPLS, et aussi aux problèmes d'interopérabilités engendrés par la coexistence au sein d'un même réseau de deux familles OAM pour la technologie MPLS-TP.

1.3. Contributions et publications

Dans cette thèse, dans sa première partie, nous avons étudié les outils OAM en vue de faciliter leur intégration au sein des réseaux de transport de nouvelle génération tel que MPLS-TP, et décortiqué les aspects problématiques liés à leur Interfonctionnement avec les OAM de réseaux MPLS existants. Ce qui a ainsi donné suite à une première publication[6].

Ensuite, nous avons proposé des solutions à ce problème se basant sur des modèles de cloisonnement et d'overlay permettant tous les deux d'assurer une continuité d'OAM de bout-en-bout, [7]. Plusieurs tests et simulations ont été effectués afin de valider nos propositions.

Dans la seconde partie, nous adoptons le paradigme Software-Defined Networking (SDN) pour proposer un prototype permettant à la fois de résoudre la problématique OAM mais également de pouvoir exécuter les OAM en tant qu'application, [8] et [9].

Enfin, nous avons étendu notre dernière approche afin de pouvoir tirer profit de la puissance des outils OAM pour optimiser les ressources réseau et pouvoir faire des décisions au niveau du « plan de contrôle » encore plus efficace. Ce qui a ainsi donné suite à une autre publication [10].

1.4. Structure de la thèse

Ce mémoire est composé de cinq chapitres dont voici un bref résumé.

Le premier chapitre présente d'abord les notions préliminaires des réseaux de transport. Nous présentons aussi les technologies de transport de nouvelle génération et qui sont basées sur Ethernet. Nous discutons ensuite des technologies IP/MPLS et MPLS-TP, ainsi que des moyens permettant d'assurer leur Interfonctionnement.

Le second chapitre décortique les différentes familles OAM de la technologie MPLS-TP. Nous commençons d'abord par mettre en évidence les similitudes et les divergences qui existent entre les familles OAM des technologies MPLS et MPLS-TP. Nous exposons les problématiques liées à leur intégration dans les réseaux existants IP/MPLS. Enfin, nous parlons des solutions qui existent déjà et qui traitent cette problématique.

Le troisième chapitre présente d'abord une première proposition se basant sur une superposition de couches réseaux en overlay pour résoudre la problématique d'interopérabilité des OAM MPLS-TP. Chacune des deux couches du modèle Overlay utilise un standard OAM diffèrent IETF ou ITU-T. En seconde partie, nous proposons un modèle de cloisonnement avec une variante utilisant un mécanisme permettant la négociation et la découverte des OAM par les différents nœuds MPLS-TP. Nous avons procédé à plusieurs tests en utilisant des équipements de transport Nokia. A la fin de ce chapitre, les différents résultats y sont ainsi présentés et analysés.

Le quatrième chapitre se penche sur la place des OAM aux seins des réseaux de transport qui se basent sur les nouvelles technologies des réseaux SDN. En effet, nous démontrerons, qu'au-delà de la résolution de tous ces problèmes d'interopérabilité et d'Interfonctionnement des OAM, il est possible de donner un nouveau sens à l'utilisation même des OAM en matière d'ingénierie de trafic, d'amélioration des performances et d'optimisation de ressources. Une première partie met en évidence l'architecture du paradigme SDN (Software Defined Network) comme moyen de fournir des services dans un réseau MPLS-TP sans pour autant modifier ses standards. En seconde partie, nous proposons une solution permettant de lancer directement les OAM comme une application ou service réseau. Nous avons pris l'exemple du service de « Mesure de délai » sur les différents chemins d'un réseau MPLS-TP. Le choix du meilleur chemin ne se fait que si le résultat du service OAM de Mesure de délai satisfait les prérequis SLA (Service Level Agreement).

Dans le cinquième chapitre, nous reprenons l'idée du chapitre précèdent pour construire une solution plus complète permettant d'utiliser les OAM afin de pouvoir optimiser les ressources d'un réseau MPLS-TP basé sur SDN. Nous proposons un modèle d'optimisation dérivé de l'algorithme de colonie de Fourmis. Nous avons alors réalisé des simulations de notre prototype et nous l'avons comparé à d'autres algorithmes. Les différents résultats y sont ainsi présentés et analysés.

Finalement, on conclut ce mémoire par une synthèse de nos contributions et de nos conclusions. Nous clôturons ce mémoire par présenter plusieurs perspectives et extensions possibles de notre contribution sur ce sujet.

Chapitre 2

Les Réseaux de Transport basés sur MPLS-TP

2.1. Introduction

Avant d'entamer l'état de l'art sur les outils d'Opérations, Administrations et Maintenances (OAM), nous estimons qu'il est judicieux de présenter et de rappeler les différentes technologies de transport. L'objectif de ce chapitre est donc de montrer les défis technologiques liés au passage de la commutation par circuit vers la commutation par paquet, et de donner des explications sur leurs principes de fonctionnement. L'état de l'art sur les OAM fera l'objet du chapitre suivant.

2.2. Introduction aux réseaux de transport

Un réseau de transport assure la transmission transparente du trafic utilisateur entre ses différents équipements en établissant et en gérant les liaisons point-à-point ou les liaisons point-à-multipoint attachés à ces équipements, indépendamment de la couche réseau supérieure qui peut exister entre ses clients. En plus du trafic client, un réseau de transport peut transporter d'autres trafics pour faciliter le fonctionnement général, tel que le trafic requis pour gérer le contrôle de connexion, la gestion du réseau et les fonctions d'opérations, d'administration et de maintenance OAM.

Le transport TDM permet aux opérateurs de fournir des connexions bidirectionnelles avec une bande passante garantie, de supporter les outils OAM, et de garantir les « accords de niveau de service » SLA. Toutefois, ce genre de technologie présente aussi beaucoup d'inconvénients. En effet, l'arrivée des Smartphones et de leurs nouveaux services associés ont presque fait tripler le trafic IP mondial entre 2013 et 2018[11], Figure 1.

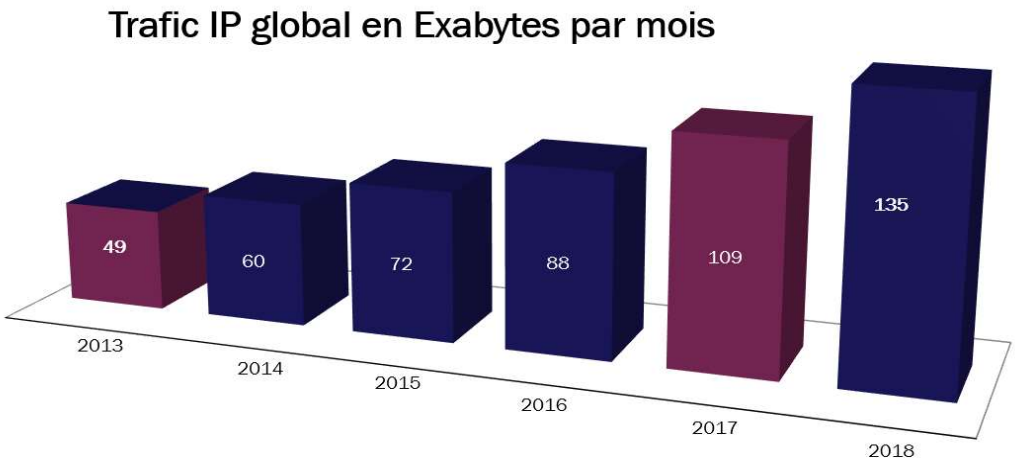


Figure 1 Trafic IP mondial entre 2013 et 2018

Une telle augmentation spectaculaire du trafic « Backhaul mobile » a surpassé les capacités des technologies TDM. Les accès circuits en T1/E1 ou en SONET/SDH sont non seulement inefficaces pour transporter le trafic natif paquets, ils sont également beaucoup plus chers (par port et par bit) que les solutions basées sur la technologie de paquets tel que Ethernet [3]. À moyen terme, préserver la technologie TDM au niveau du transport est intenable pour les opérateurs. La solution a été de définir un ensemble de protocoles et procédures qui héritent de toutes les performances des réseaux à technologie de paquets tout en restant conformes aux exigences OAM du monde des transports. La Figure 2 (a) illustre bien comment la rénovation dans les réseaux de transport permet de réduire les différents coûts d'investissements ou d'exploitation, et elle montre aussi plus de service engendrerait plus de revenus et qu'à partir d'un certain seuil d'investissement le coût de bande passante reviendrait moins cher. Tandis que la Figure 2 (b) résume les motivations pour créer une nouvelle technologie de transport supportant à la fois les accès circuit et paquet.

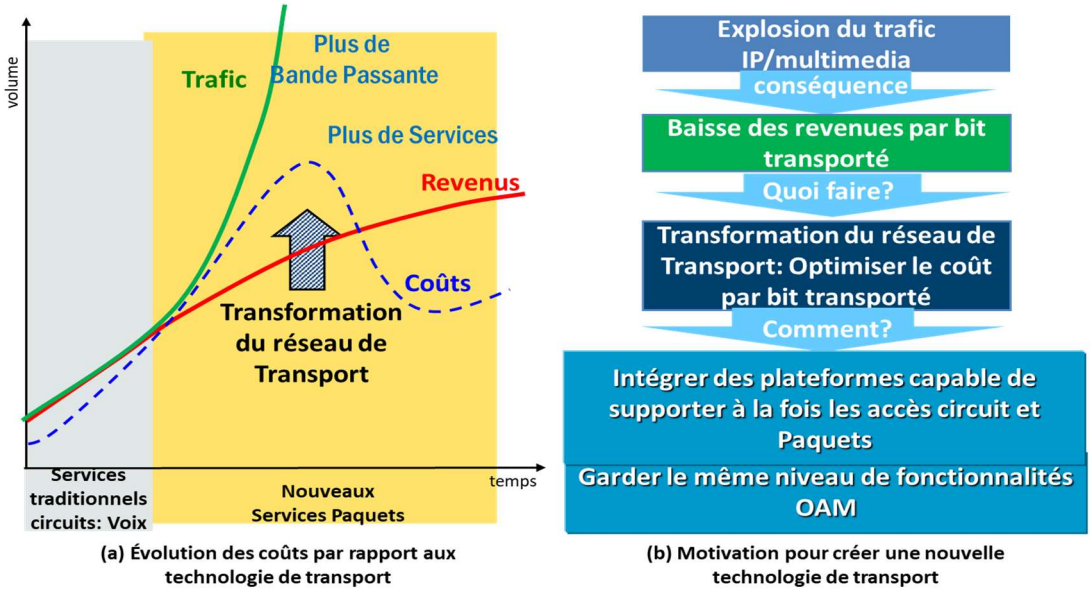


Figure 2 Les tendances des revenus des opérateurs par type de service

Les « réseaux nouvelle génération » se caractérisent par les concepts suivants [3]:

- La commutation par paquet est la seule base du réseau de transport ;
- Les caractéristiques des services doivent être indépendantes de la technologie de transport sous-jacente ;
- La technologie d'accès doit être transparente pour accéder à un service offert par le réseau de transport ;
- Une QoS doit être fournie de bout-en-bout avec une garantie comparable aux technologies de transport traditionnelles ;

Il existe plusieurs facteurs qui motivent les opérateurs (possédant des réseaux de transport) à faire la transition depuis les technologies de transport orientées circuit vers d'autres technologies de transport orientées paquet. En effet, le principal vecteur de cette transition est celui du développement des technologies mobiles en termes d'interface d'accès. Ces dernières se basaient sur des accès E1 (2048 kbit/s) ou T1 (1544 kbit/s) pour les stations de base 2G, tandis que maintenant les NodeB du 3G ou encore les eNodeB du LTE disposent d'interfaces en GigaEthernet. La Figure 3 illustre bien ce passage du mode par circuit au mode par paquet dans les technologies de transport dans le monde du réseau mobile.

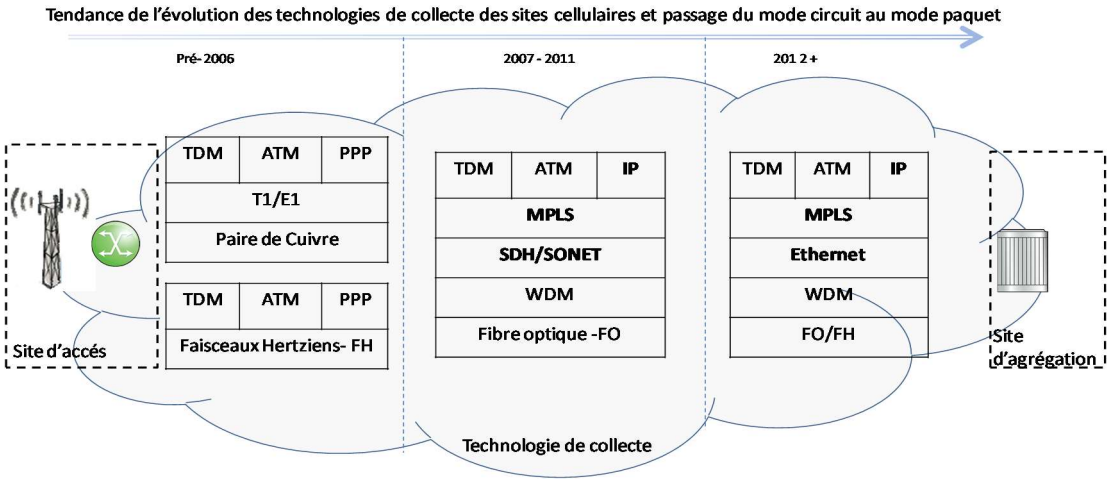


Figure 3 Passage du mode circuit au mode paquet avec la technologie Ethernet dans les réseaux mobiles

Les études du marché mondial des télécommunications démontrent que le passage vers l'Ethernet se fait d'une manière spectaculaire pour passer de 23% en 2013 à 76% en 2018 comme le montre la Figure 4, [12].

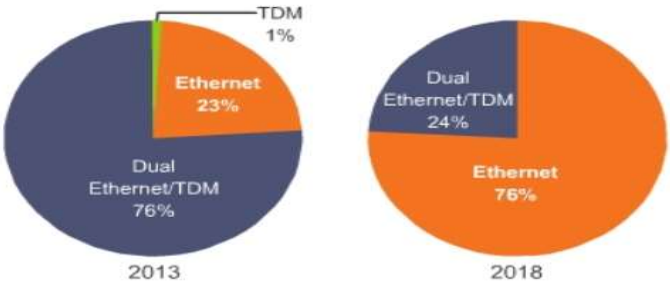


Figure 4 Évolution de la transition des équipements TDM vers l'Ethernet

Les sections suivantes présentent les avantages des technologies de transport Ethernet.

2.2.1. Granularité de la bande passante et flexibilité des débits

Les technologies traditionnelles de transport TDM supportent une bande passante fixe sans aucun multiplexage statistique. La bande passante est une ressource réservée tout au long du réseau de transport, indépendamment de son utilisation effective ou pas par le client. En revanche, les technologies de paquets soutiennent le multiplexage statistique, ce qui est considéré comme la motivation la plus importante pour la transition des technologies de transport traditionnelles vers des technologies de transport de paquets.

En plus, la prolifération des nouvelles applications distribuées, qui communiquent souvent en mode rafale (burst) avec d'autres serveurs sur le réseau, a été le moteur de l'adoption des techniques de transport de paquets. En effet, le multiplexage des paquets d'un trafic en rafale, en provenance d'une source, permet une utilisation plus efficace de la bande passante que celui dans les technologies TDM traditionnelles en mode circuit.

La non granularité des débits des connexions de données, dans les technologies de transport traditionnelles, est limitée à cause de la hiérarchie rigide de la technologie PDH (Plesiochronous Digital Hierarchy) avec des valeurs de débits figées: à 1.544Mbps (DS-1 : Digital Signalling), ou à 44.736Mbps (DS-3). On constatera la même problématique de débits au niveau de la technologie SONET (Synchronous Optical NETworks) avec des valeurs de débits figés comme par exemple : OC3 à 155 Mbps (Optical Contener) ou OC12 à 622.08 Mbps.

D'un autre côté, les technologies de transport par paquets offrent de la granularité de débits et ont l'avantage de pouvoir fournir des connexions de données à des débits très flexibles. Cette flexibilité en termes d'offre de bande passante, dans les technologies de transport paquet, est particulièrement importante pour les applications d'aujourd'hui dont le profil et les caractéristiques varient considérablement d'une application à une autre. Les opérateurs de transport profitent de cette flexibilité de débits pour mutualiser les ressources (liens et ports physique) et également pour proposer des offres sur mesures répondant d'une manière précise aux besoins de leurs clients.

2.2.2. Qualité de service QoS

Les technologies de transport traditionnelles (tels que TDM) fournissent une garantie de bande passante, mais sont incapables de reconnaitre le type de trafic transporté. Donc, elles ne sont pas en mesure de fournir une qualité de service selon le type de trafic au sein d'un même accès TDM par exemple. A l'inverse des réseaux de transport de paquets qui sont capables de classifier le trafic selon la politique de QoS définie et éviter ainsi une dégradation de service en cas de congestion.

Par ailleurs, malgré l'expansion de la technologie Ethernet dans le monde des connexions longue distance WAN Ethernet, il en reste que la gestion de ces connexions est toujours un défi. Cette technologie a été conçue initialement comme une technologie de réseau local (LAN). Elle ne disposait pas de mécanismes OAM associés aux technologies de transport

de classe transporteur tels que SDH ou réseau de transport optique (OTN) qui permettent de surveiller et de maintenir les liens réseau. La technologie Ethernet classique souffre de certaines limitations :

- Evolutivité (limitation à 4096 VLAN),
- Restauration très lente (2 secondes à 120 secondes avec Spanning Tree et ses dérivés),
- QoS limitée (les queues sont définies saut par saut),
- Ingénierie de trafic très limitée.

Ainsi pour que la technologie Ethernet soit considérée comme technologie de transport, il faut qu'elle dispose des fonctionnalités de transport telles que l'architecture de la couche réseau, la séparation entre les clients et les outils de gestion qui permettent aux transporteurs de faire des déploiements à grande échelle. Ainsi, avec l'avènement du PBB-TE et du MPLS-TP, il est maintenant possible d'utiliser Ethernet comme une technologie de transport pour les réseaux nouvelle génération.

Dans la suite de ce chapitre, nous exposons en détail les principaux protocoles utilisés dans les réseaux de transport Ethernet PBB-TE (Provider Backbone Bridges Traffic Engineering) et MPLS-TP et qui sont basés nativement sur la technologie « tout paquet ».

2.3. Les réseaux de transport Ethernet (Carrier Ethernet)

La technologie Ethernet est très largement déployée dans les réseaux locaux (LAN) et les réseaux d'accès ADSL (Asymmetric Digital Subscriber Line) : en mode IP sur Ethernet. L'évolution de la technologie Ethernet, vers la transmission full-duplex, l'auto négociation et la suppression de la technique de détection de collision CSMA/CD (Carrier Sense Multiple Access with Collision Detection), a favorisé son introduction progressive dans les réseaux capillaires et les réseaux de collecte.

2.3.1. Définitions et caractéristiques

Les termes réseau de transport Ethernet ou Carrier Ethernet ou Ethernet de classe opérateur font référence à l'utilisation de la technologie Ethernet et de toutes ses extensions techniques comme technique de transport dans les réseaux globaux (WAN).

Afin de permettre à la technologie Ethernet d'être utilisée comme technologie de transport, elle doit être en mesure de fournir les caractéristiques désirées d'un réseau de classe opérateur. Le Metro Ethernet Forum (MEF) a développé le concept de «Carrier Ethernet» qui permet d'offrir un service omniprésent de classe opérateur, caractérisé par cinq attributs lui permettant de se différencier de l'Ethernet des LAN classiques [13]. Le terme « Carrier Ethernet » désigne « un réseau de classe transporteur ubiquitaire standardisé offrant du service ». Ils existent cinq attributs qui permettent de distinguer un réseau « Carrier Ethernet » par rapport à un réseau LAN Ethernet privé [14]:

- Services standardisés : les Services « Liaison privée virtuelle » E-LINE (Ethernet Line) et « LAN privé virtuel » E-LAN (Ethernet LAN) doivent être fournis en toute transparence pour les topologies point-à-point et (multi)point-to-multipoint.
- Evolutivité : la technologie de transport doit être capable d'opérer dans des réseaux larges et complexe, et à tous les niveaux hiérarchiques du réseau : métro, accès ou cœur.
- Fiabilité : le réseau de transport doit être en mesure de réagir aux disfonctionnements à l'aide de mécanismes de calcul et de convergence de chemins dans un intervalle inférieur à 50 ms.
- Qualité de service: le réseau de transport doit être capable de satisfaire les exigences spécifiées dans les accords de service avec le client SLA (Service Level Agreement). Ces contraintes SLA sont en rapport avec les paramètres du réseau bande passante, délai, gigue et d'autres.
- Gestion des services: Les opérateurs doivent être capables de superviser et de diagnostiquer le réseau de transport à l'aide d'outils OAM adaptés.

Le Transport Ethernet est donc un service de transport de trames Ethernet disposant de certains attributs qui permet la création d'un réseau Ethernet étendu. Pourtant, dans cette définition, la technologie utilisée pour le transport des trames Ethernet n'est pas forcément celle définie par les normes de la famille IEEE 802.1/802.3 puisque les services de transport de trames Ethernet peuvent utiliser aussi des technologies de transport de longue portée.

D'un point de vue standards, le IEEE a normalisé l'Ethernet, pour développer les technologies «Carrier Ethernet», afin qu'elle soit utilisée entant que technologie de transport dans l'accès, l'agrégation et plus récemment dans le cœur du réseau, à partir des technologies déjà existantes. Une série de normes a été élaborée, se focalisant essentiellement sur les mécanismes permettant à l'Ethernet d'être utilisé dans les réseaux de transport. Ceci est devenu possible grâce à l'utilisation des mécanismes existants tel que: la gestion des classes de service des VLAN avec le champ 802.1p de la norme IEEE 802.1q, les mécanismes comme le « Q-in-Q » et le « MAC-in-MAC », les technologies Provider Bridges 802.1ad [15] et PBB (Provider Backbone Bridges) 802.1ah [16].

Le Metro Ethernet Forum (MEF) a publié de nombreuses spécifications techniques décrivant les attributs d'interface et de service Ethernet pour accélérer le développement de cette technologie. D'autres organisations comme l'IEEE, l'IETF et l'ITU-T ont continué sur cet élan pour développer en 2009 la première norme Ethernet orientée connexion qui est la technologie PBB-TE (Provider Backbone Bridge – Traffic Engineering) dans le standard 802.1Qay [17]. Cette technologie constitue une évolution du standard existant PBB IEEE 802.1ah plus connu sous le nom de « MAC-in-MAC », et se base sur une approche hiérarchique de VLAN. Par ailleurs, le deuxième candidat pour être une technologie Carrier Ethernet est le MPLS Transport Profile (MPLS-TP) qui prend sa puissance du protocole MPLS mature et largement déployé chez les opérateurs. La Figure 5 trace cette évolution technologique où chaque organisme essaie d'occuper la place des standards.

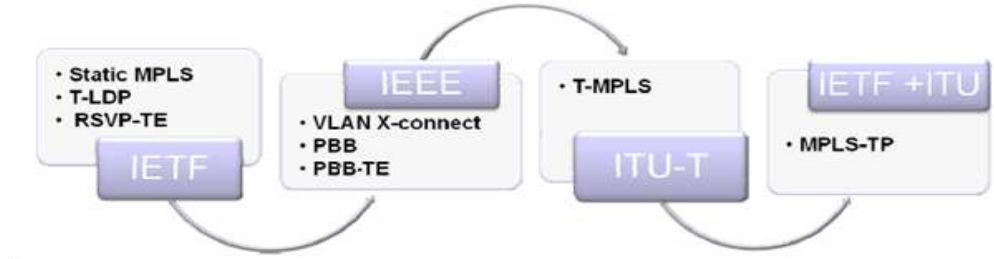


Figure 5 Evolution des technologies selon les organismes de standardisation

T-LDP: Targeted Label Distribution Protocol

RSVP-TE: Resource Reservation Protocol - Traffic Engineering

T-MPLS: Transport MPLS

Dans les sous sections suivantes, on exposera brièvement la technologie PBB-TE et on traitera avec plus de détails MPLS-TP en essayant de montrer pourquoi cette dernière a eu plus de succès chez les opérateurs et les équipementiers.

2.3.2. Réseaux de transport de nouvelles génération

Il existe deux principale technologies de transport basée sur l'Ethernet : PBB-TE et MPLS-TP[18].

La technologie Provider Backbone Bridges (PBB-TE), définit par la norme 802.1Qay [4], prend ses origines de la norme VLAN 802.1q [19] qui constitue un point de démarrage pour notre compréhension de cette nouvelle technologie.

La technologie PBB-TE est essentiellement basée sur le concept de transférer le trafic via des tunnels ou chemins Ethernet commutés (Ethernet Switched Paths ESP) qui sont préconfigurés par un système de gestion réseau centralisé. Le plan de contrôle du PBB-TE se base sur l'utilisation du GMPLS (Generalized MPLS)[20] mais la majorité des implémentations actuelles PBB-TE comptent sur des solutions de Systèmes de Gestion du Réseau (Network Management System NMS) statiques [21].

Dans PBB-TE, les adresses MAC backbone, B-DA (Backbone Destination Address) et B-SA (Backbone Source Address), associées au tag VLAN sont utilisées pour définir explicitement le port (ou les ports) de sortie de la trame Ethernet, Figure 6. Ce concept peut alors être appliqué aux différents types de service Ethernet : E-LAN, E-Tree et E-line.

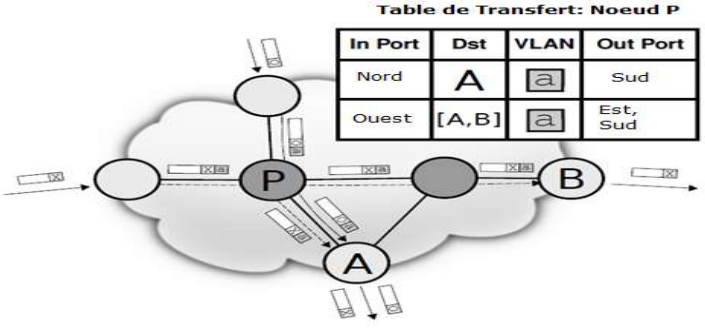


Figure 6 PBB-TE : Exemple de transfert de trame Ethernet

La technologie PBB-TE se diffère de PBB dans certains aspects en abandonnant certaines fonctionnalités, comme: l'apprentissage des adresses MAC, le Spanning Tree, ou l'inondation des trames inconnues, elle a introduit l'usage d'un système de gestion NMS pour permettre le calcul des tunnels qui sont signalés dans des tables de transfert. Cela permet aussi le calcul des chemins de protection et d'autres opérations liées à l'ingénierie du trafic. La technologie PBB-TE peut être utilisée en conjonction avec des mécanismes OAM tel que 802.3ah ou CFM (Connectivity Fault Management) pour fournir la résilience exigée par les réseaux de transport.

Toutefois, la technologie PBB-TE présente des limitations comme par exemple le fait qu'elle est restreinte à un seul domaine opérateur, et qu'elle ne soit pas très adaptée aux architectures maillées des nouvelles technologies radio comme 4G/LTE (Long Term Evolution).

La deuxième technologie de transport la plus utilisée est MPLS-TP. Cette technologie découle historiquement de la technologie T-MPLS (Transport MPLS) qui a été définie au sein de l'ITU-T comme technologie de transport par paquets orientée connexion et qui avait comme objectif de définir un profil transport au protocole MPLS en réduisant la complexité du plan de données et en dissociant strictement le plan de contrôle du plan de données. Le plan de contrôle de la technologie T-MPLS est optionnel et les fonctions typiques du plan de contrôle de MPLS comme la protection et l'OAM sont réalisées au niveau du plan de données. Un des objectifs de la technologie T-MPLS était de s'affranchir des protocoles de routage afin de réduire les coûts. L'ITU-T a standardisé cette technologie par une série de recommandations [22]. Les deux organisations ITU-T et IETF ont collaboré à l'extension des protocoles MPLS afin de satisfaire l'ensemble des attentes d'un réseau de transport, et d'abandonner le développement de la technologie T-MPLS pour donner naissance à MPLS-TP (MPLS – Transport Profile). Les principes de MPLS-TP sont les mêmes que pour T-MPLS sauf que la définition des protocoles est prise en charge par l'IETF.

Dans la suite, on fera un rappel des principales composantes de la technologie IP/MPLS afin d'appréhender avec aisance le concept de la technologie MPLS-TP.

2.3.3. Généralités sur la technologie MPLS

La technologie Multi Protocol Label Switching (MPLS) est une norme définie par l'IETF et qui permet d'acheminer les paquets en se basant sur la commutation de label [23]. Cette commutation part d'un principe de regrouper chaque ensemble de paquets semblables selon certaines caractéristiques constituant des classes nommées FEC (Forwarding Equivalence Class). Ces classes sont identifiées par certains critères qui peuvent être: paquets ayant le même préfixe IP source ou destination, paquets identifiant une même application, qualité de service demandée, etc.

La norme MPLS se caractérise par les propriétés suivantes :

- Spécification des techniques et mécanismes permettant le transport des paquets IP avec diverses granularités des flots entre deux points, deux machines ou deux applications.

- Séparation des niveaux paquet et trames.
- Mise en relation de l'adresse IP du destinataire avec une référence d'entrée dans le réseau.
- Support de protocoles de routage comme OSPF (Open Shortest Path First) et de signalisation comme RSVP par les nœuds d'accès au réseau MPLS.
- Utilisation de différents types de trames.

a) Fonctionnement d'un réseau MPLS

L'acheminement du trafic se fait via des chemins préétablis appelés LSP (Label Switched Path). Les nœuds se trouvant aux bords d'un réseau IP/MPLS s'appellent LER (Label Edge Router) ou PE (Provider Edge). Les LER ont comme principale mission d'ajouter ou de supprimer les Labels (étiquettes) aux paquets IP à l'entrée ou à la sortie d'un réseau IP/MPLS. Les routeurs intermédiaires se trouvant à l'intérieur du réseau et qui ont pour rôle de commuter les paquets en fonction de leurs labels s'appellent LSR (Label Switched Router). Ainsi, un LSP représente une succession de labels qui affecte un paquet transitant depuis un LER vers un autre LER. On notera aussi que chaque LER ou LSR attribue un label à une FEC (Forwarding Equivalence Class) indépendamment de ses voisins. Ces associations FEC/Labels sont gérées et distribuées entre les différents LER et/ou LSR via des protocoles de distribution de Labels comme LDP ou RSVP-TE. Un LSR possède, donc, des tables de commutations qui indique les références associées aux FEC et qui sont utilisées pour sélectionner le chemin LSP adéquat pour chaque paquet. Ces tables de commutations maintiennent la correspondance entre les couples (Interface d'entrée, ancien Label) et (Interface de sortie, Nouveau Label). Ainsi, l'ensemble des trames d'une même FEC sont envoyées sur la même interface de sortie. Cette table de commutation est appelée LFIB (Label Forwarding Information Base) comme indiqué dans la Figure 7.

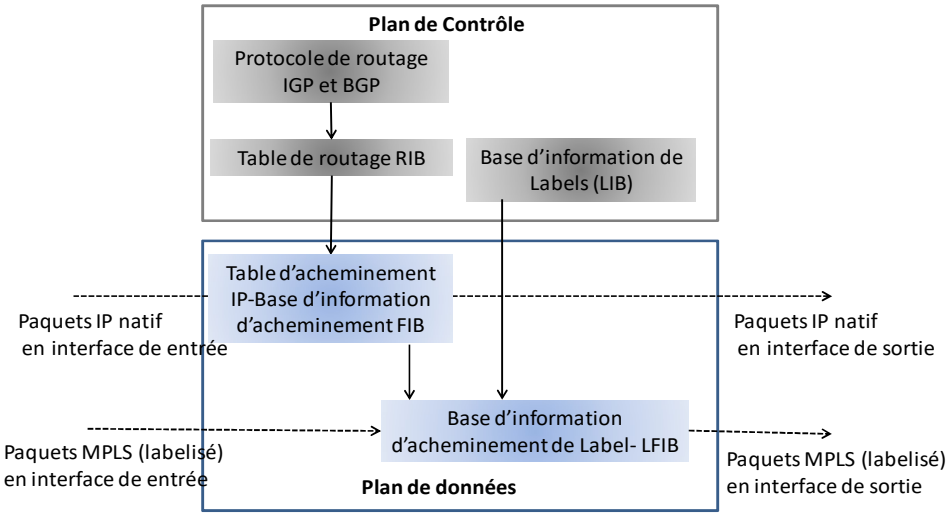


Figure 7 LER/LSR : interaction entre les différentes composantes MPLS

b) Les applications MPLS

La technologie MPLS, boostée par son concept de commutation par label, se caractérise par une vitesse de commutation et de transfert de données meilleur que celle utilisée par une technologie basée sur IP qui s'appuie sur la vérification du champ IP destination pour l'acheminement de paquets [24]. Ceci a permis de développer plusieurs types de service autour du MPLS comme :

- Réseau Privé virtuel niveau 3 L3VPN (Layer 3 Virtual Private Network): définit par le RFC4364 [25]
- Réseau local privé virtuel VPLS: qui est un service MPLS VPN niveau 2 [26]
- Le PW (Pseudowire) appelé aussi PWE3 (Pseudowire Edge-to-Edge Emulation): a pour but d'émuler à travers un réseau de paquets un service natif des couches 1 ou 2 du modèle OSI [27]. Plusieurs types de PWE3 existent et sont présentés dans le Tableau 1.

Service émulé	RFC	Description
Ethernet	4448	Encapsulation de l'Ethernet sur MPLS
TDM	4553	Transport du TDM structuré sur un réseau paquet
PPP/HDLC	4618	Encapsulation du PPP/HDLC sur MPLS
Relai de trame	4619	Encapsulation du Relai de trame sur MPLS
ATM	4717	Encapsulation de l'ATM sur MPLS
SONET/SDH	4842	Emulation de circuit SONET/SDH un réseau paquet
TDM	6086	Transport du TDM non-structuré sur un réseau paquet

Tableau 1 Les encapsulations standards des services émulés dans MPLS

c) La Qualité de service (QoS)

La technologie MPLS permet d'assurer une QoS de bout en bout, pour tous les services cités auparavant, en prenant en considérations quatre principaux paramètres [28]: le délai de transmission, la gigue, la bande passante et le taux de perte. Deux modèles QoS existent: services intégrés IntServ (IETF RFC 1633 1994) et services différenciés DiffServ (IETF RFC 2475 1998).

d) Ingénierie de trafic

MPLS dispose de mécanisme d'ingénierie de trafic lui donnant la possibilité de pré-calculer des LSP soumis à certaines contraintes. Ce pré-calcul de chemin est possible grâce à certains protocoles de réservation de ressources comme RSVP ou LDP. Des LSP sont alors soumis

à certaines contraintes. Certains attributs, décrits dans le Tableau 2, sont utilisés pour contrôler les ressources liées aux chemins associés au LSP.

Attribut	Description
Bande passante	Spécifie la bande passante minimale à réserver
Nature du chemin	Caractérise si le chemin est manuel ou dynamique
Priorité	Priorité du LSP par rapport aux autres LSP en termes de réservation de ressource
Couleur	Valeur administrative
Optimisation	Permet de toujours choisir le chemin optimal
Reroutage	Reroutage en cas de panne

Tableau 2 Attributs des ressources d'un LSP

Les algorithmes de calcul de chemin basés sur les contraintes, généralement résultant des termes d'un contrat de niveau de service (SLA), utilisent les attributs comme « bande passante » ou « couleur » pour forcer le LSP à prendre le bon chemin.

e) Les OAM MPLS

Les OAM MPLS [31] sont des mécanismes, utilisés au niveau de la couche MPLS, qui permettent de détecter, d'identifier et de localiser les défauts au niveau du plan data d'un réseau MPLS.

Les OAM MPLS peuvent opérer au niveau du tunnel de transport LSP comme le protocole BFD (Bidirectional Forwarding Detection), le ping et le traceroute LSP:

- BFD [32]: Il a pour objectif de vérifier la continuité, par l'envoi à intervalle régulier de paquets « Hello », et ainsi détecter les pannes entre deux équipements adjacents. Ceci permettra de pouvoir déclencher les mécanismes de protection dans des délais inferieur à 30 millisecondes pour certaines implémentations matérielles. BFD est un protocole qui fonctionne en mode point-à-point bidirectionnel. BFD est souvent utilisé pour déclencher d'autres mécanismes de protection de lien ou de nœud de réseau tel que le Fast ReRoute.
- LSP Ping/Traceroute : Le principe du LSP Ping est de vérifier si le LSP fonctionne dans les deux directions tandis que le « LSP Traceroute » permet d'identifier le chemin emprunté par le LSP [33].

D'autres OAM MPLS peuvent également opérer au niveau du tunnel de service PseudoWire comme les VCCV (Virtual Circuit Connectivity Verification) et certains messages LDP:

- LDP : L'envoi de messages de notification LDP (LDP Label Withdraw) au niveau d'une session Target-LDP du PseudoWire permet d'informer le PE distant d'un défaut sur la connectivité du tunnel de service ainsi qu'une information sur la nature du défaut [34].

- VCCV: L'utilisation du canal VCCV permet de tester la connectivité et de détecter les défauts sur un PW. Ce même canal peut également notifier au PE distant un défaut au niveau du circuit d'attachement local (AC). Ce canal permet d'utiliser les protocoles tels que BFD ou LSP Ping au niveau du tunnel de service [35].

2.3.4. Les principes de la technologie MPLS-TP

On peut définir la technologie MPLS-TP comme étant un sous-ensemble de la technologie MPLS à laquelle nous avons rajouté des extensions compatibles avec les exigences des réseaux de transport. Les domaines clés dans la standardisation de MPLS-TP sont : le transfert de données, les OAM, la robustesse, le contrôle et la gestion [36], Figure 8.

La technologie MPLS-TP possède les caractéristiques clés suivantes:

- Orientée connexion: pour y parvenir les techniques ECMP (Equal Cost Muti-Path) et MP2P (Multi Point to Point) y ont été exclues, et le PHP (Penultimate Hop Popping) y est désactivé par défaut, [23];
- Le client est agnostique à la couche physique: permettant aux paquets MPLS d'être livrés sur une variété d'infrastructures physiques comme l'Ethernet en utilisant des technologies comme WDM (Wavelength Division Multiplexing)
- Plan de Contrôle: statique ou dynamique;
- Un ensemble riche de fonctions OAM similaires à celles disponibles dans les réseaux de transport optique existants (par exemple, SONET / SDH, OTN); ces fonctions OAM appartiennent au plan de données MPLS-TP et sont indépendantes du plan de contrôle;
- des mécanismes de protection de chemin et le mécanisme basé sur le plan de contrôle;
- L'utilisation du G-ACh (Generic Associated Channel) qui permet de supporter les fonctions FCAPS (Défaut, Configuration, Administration, Performance et sécurité). Le G-ACh fournit un canal de données logique auxiliaire associé à un LSP, un PW ou une section (un lien) sur lequel peuvent circulé divers protocoles.
- La configuration des éléments du réseau via un système de gestion centralisé NMS et/ou un plan de contrôle distribué.

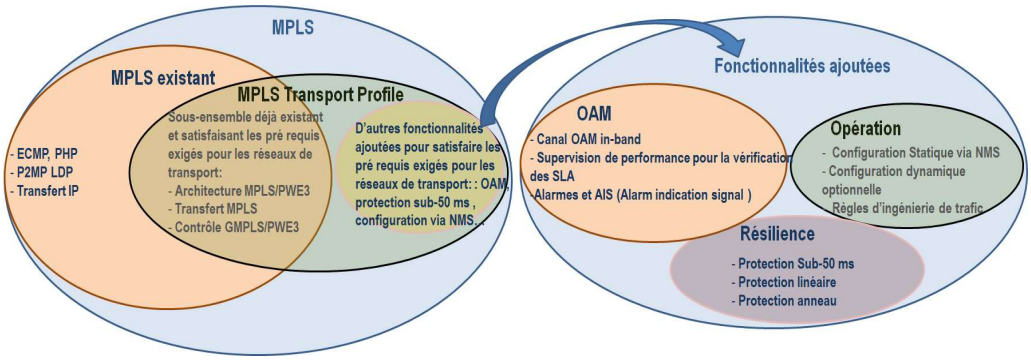


Figure 8 MPLS-TP: un mélange de sous-ensemble de MPLS et de nouvelles fonctionnalités orientées transport

La technologie MPLS-TP est une solution basée sur les standards existant du protocole MPLS décrivant les PW et LSP. En effet, les LSP sont utilisés pour assurer le transport et les PW sont réservés à la couche client ou service (PW mono ou multi segment), Figure 9 et Figure 10 [37]. On peut voir aussi dans ces deux figures que la couche client du protocole MPLS-TP peut être aussi de n'importe quel type.

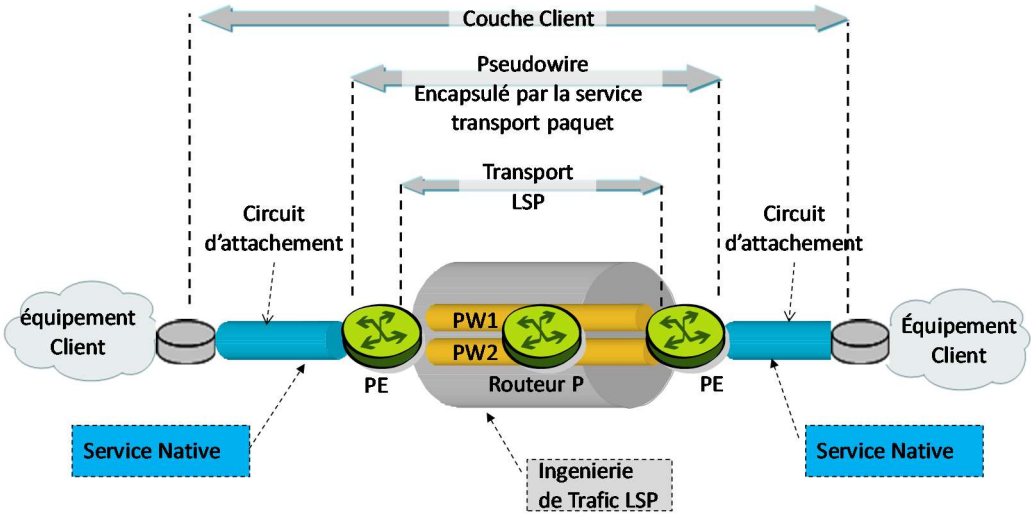


Figure 9 Architecture MPLS-TP: multiple Pseudowire

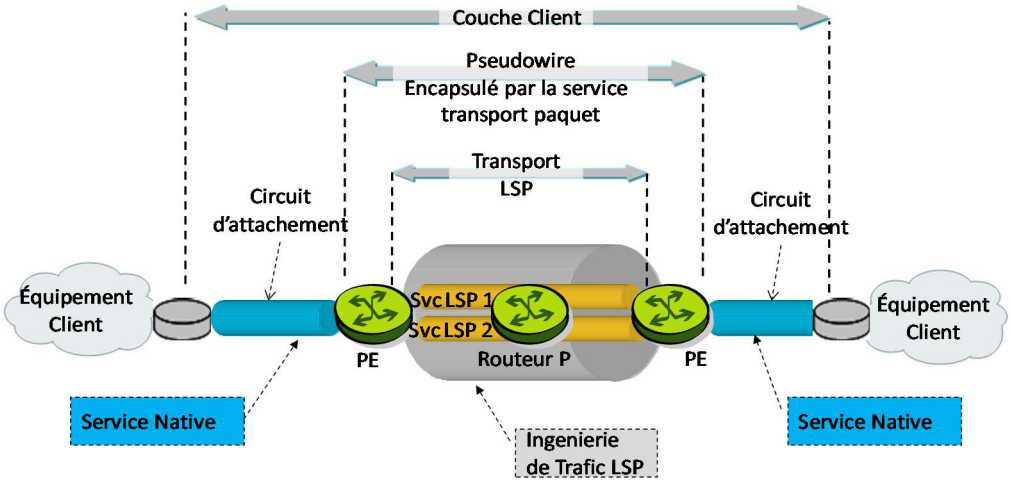


Figure 10 Architecture MPLS-TP: multiple tunnels transportés

La technologie MPLS-TP supporte deux mécanismes d'adaptation pour transporter les services natifs:

- Le pseudowire PW: qui permet d'émuler certains services comme l'Ethernet, le Frame Relay, le PPP ou encore le High-Level Data Link Control (HDLC). Ces fonctions d'adaptation sont encapsulées comme une charge utile (payload);
- Le LSP : qui permet de fournir l'adaptation pour tout type de service natif comme le trafic IP ou encore MPLS (exemple: PW sur LSP, ou IP sur LSP). La fonction d'adaptation utilise le même format d'encapsulation que le protocole MPLS.

Outre les avantages que fournissent ces deux mécanismes d'adaptation en matière de flexibilité et de diversité de type de service pouvant être transporté, MPLS-TP dispose de quatre composants essentiels, Figure 11 :

- Un Plan de données: qui est resté d'ailleurs le même que celui de la technologie MPLS pour ainsi faciliter l'Interfonctionnement entre MPLS-TP et MPLS ;
- Un Plan de contrôle: qui est optionnel, et qui peut être soit dynamique en réutilisant GMPLS soit statique via la plate-forme de gestion;
- OAM: avec des OAM au même niveau d'exigence que celles utilisées dans les réseaux SDH. C'est une condition nécessaire pour que MPLS-TP soit considérée comme une technologie de transport ;
- La protection et la résilience: avec un mode opératoire égale à celui des réseaux SDH;

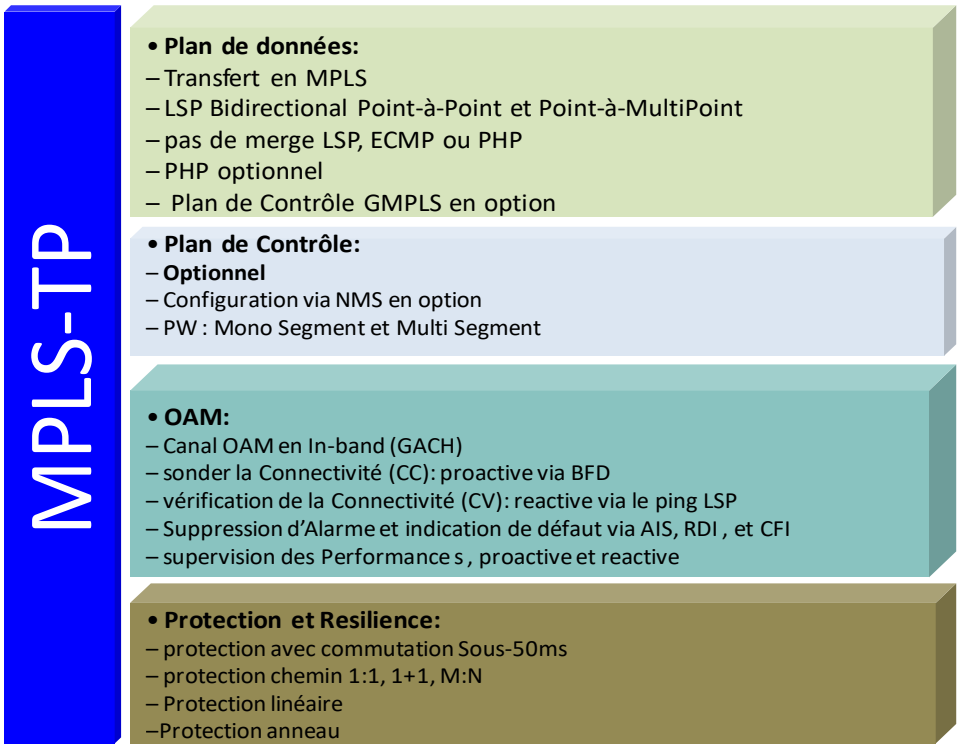


Figure 11 MPLS-TP : Les attributs majeurs

Nous présentons, ci-dessous, en détail les quatre composants majeurs de la technologie MPLS-TP.

a) Plan de données

Le plan de données est l'un des éléments les plus importants et qui est commun entre les technologies MPLS et MPLS-TP. Il utilise la même architecture de transfert que le standard MPLS et il est totalement séparé du plan de contrôle. Il existe, tout de même, quelques fonctionnalités qui ont été abandonnées par MPLS-TP comme PHP, la fusion des Label-Switched Paths (LSP), et les multi-chemins à coût égaux (ECMP).

Le traitement des PW mono et multi segments au niveau MPLS-TP est fait de la même manière que celui de la technologie MPLS sans modifications ni extensions [38]. On citera quelques exemples des procédures supportées par le plan de données, Figure 12:

- Le « control word » du Pseudowire Emulation Edge-to-Edge (PWE3) pour une utilisation sur un réseau MPLS de bout-en-bout;
- Méthode d'encapsulation pour le transport de l'Ethernet sur les réseaux MPLS
- Structure-Agnostique par rapport à la technologie TDM via des paquets de type SAToP (Structure-Agnostic TDM over Packet);
- Méthodes d'encapsulation pour le transport des PPP (Point to Point Protocol) et HDLC (High-level Data Link Protocol) sur des réseaux MPLS
- Méthodes d'encapsulation pour le transport de Frame Relay sur des réseaux MPLS
- Méthodes d'encapsulation pour le transport de l'ATM sur des réseaux MPLS
- Un service ATM transparent au niveau des Pseudowire End to End Emulation (PWE3);
- Emulation des circuits SONET/SDH sur un réseau commuté par paquets (CEP : Circuit Emulation over Packet);
- Emulation d'un service Structure-Aware TDM Circuit Emulation sur réseau commuté par paquets (CESoPSN);
- TDM sur IP (TDMoIP);
- Méthodes d'encapsulation pour le transport de trames « Fiber Channel » sur les réseaux MPLS ;

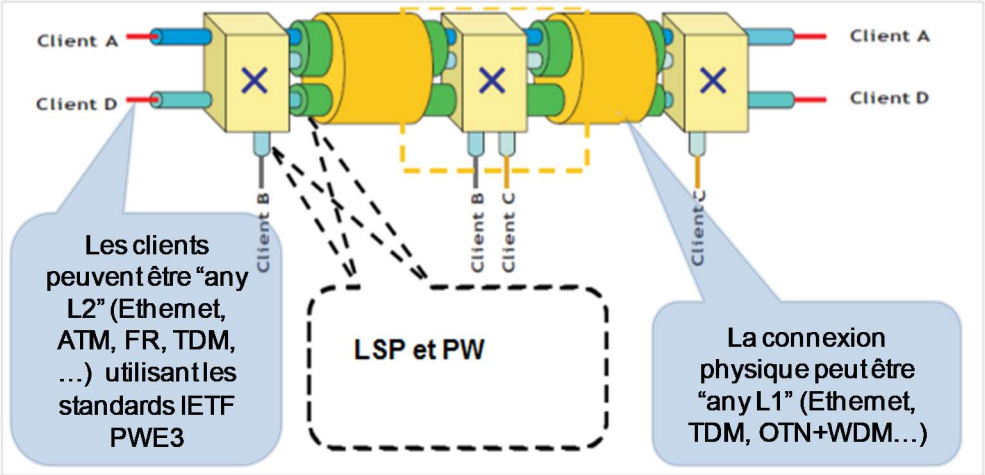


Figure 12 Plan de données MPLS-TP : procédures supportées

Certaines exceptions existent lorsque l'on compare les deux technologies MPLS et MPLS-TP notamment le mode de transfert des paquets OAM. La technologie MPLS utilise le plan de contrôle pour envoyer les paquets OAM alors que la technologie MPLS-TP utilise le plan de donnée pour le faire d'où le terme « in-band » associé aux OAM MPLS-TP.

Une autre différence majeure existe et qui concerne le mode de transfert unidirectionnel ou bidirectionnel au niveau LSP. En effet, MPLS est basé sur le paradigme traditionnel de routage IP: le trafic de A à B peut circuler sur un chemin différent que celui de B à A. Les LSP bidirectionnels co-routés sont supportés par MPLS-TP et sont définis en associant les directions amont et aval pour suivre le même chemin: même nœuds et mêmes liens. En plus, MPLS-TP supporte aussi les LSP unidirectionnel point-à-point et point-à-multipoint.

b) Plan de contrôle

La fonction majeure du plan de contrôle est de dicter comment le transfert sera fait, en l'occurrence pour un réseau MPLS-TP, comment le LSP sera configuré ou décrit. Il est facultatif et le modèle statique peut être tout simplement utilisé via une station de gestion NMS pour configurer les PW et LSP, comme c'est déjà le cas des réseaux de transport classiques où aucun protocole de routage ni de plan d'adressage IP n'est pris en compte pour définir les circuits de bout en bout. Le plan de contrôle est également totalement séparé du plan de données, et tout manquement à cette partie ne devrait pas affecter le mode de transfert [39].

Le plan de contrôle peut également être dynamique, Figure 13. Les LSP sont configurés et établis en utilisant la suite GMPLS qui est déjà mature dans le monde MPLS [20]. Le signalement des labels de services PW est aussi fait d'une manière dynamique à l'aide du protocole LDP « ciblé » T-LDP [40]. En voici quelques avantages:

- Le GMPLS supporte l'ingénierie du trafic, les mécanismes de qualité de service et l'utilisation efficace des ressources, les mécanismes globaux pour la protection et la restauration rapide, la séparation entre les canaux de contrôle et de données garantissant que la défaillance de l'un ne porte pas atteinte de l'autre ...
- Le T-LDP permet l'évolutivité des services, la signalisation des labels PW et sa correspondance aux classes de transfert équivalente « Forwarding Equivalence Class » (FEC), le support des mono et multi segment PW, l'utilisation d'adressage évolué des PW avec MPLS-TP...

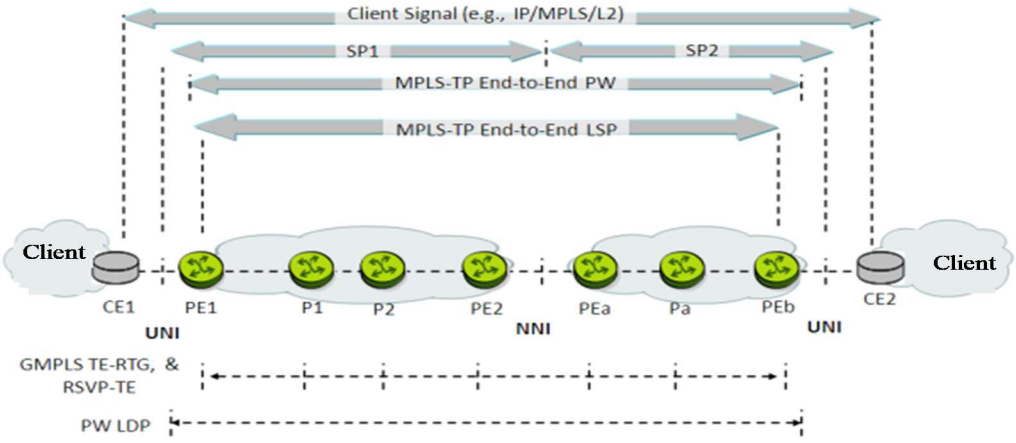


Figure 13 Le plan de contrôle dynamique du MPLS-TP

Le plan de contrôle dynamique est certes évolutif et peut fournir plusieurs fonctions intéressantes, comme la signalisation, le routage et l'ingénierie de trafic, mais l'option statique reste la plus adoptée par les opérateurs dont le personnel, en charge de la partie

transport, est déjà habitué à ce mode de plan contrôle, statique via NMS, avec les réseaux de transport traditionnels.

c) Protection et résilience

Conformément aux exigences de la technologie MPLS-TP, la protection et la résilience sont des caractéristiques vitales des réseaux de transport Ethernet. Les mécanismes de protection de commutation sont capables de fournir des temps de restauration inférieur à 50 ms [41]. Les mécanismes de protection et de restauration proposés sont basés sur le plan de donnée où les OAM sont in-band (in-band = incorporée avec le trafic utilisateur) afin de permettre d'optimiser l'utilisation des ressources réseau. Il existe, alors, plusieurs modèles de protection, Figure 14:

- La protection linéaire 1+1: Un pont permanent envoie continuellement du trafic à la fois sur le chemin actif et sur le chemin de détour, et c'est le rôle du pont « sélecteur » de choisir son chemin actif. Un protocole de coordination des états PSC (Protection State Coordination Protocol) est utilisé au niveau du chemin de détour (via le G-ACh) dans le cas où les flux sont bidirectionnels afin de permettre la coordination des deux extrémités;
- La protection linéaire 1:n (incluant 1 :1): Un pont « sélecteur », au niveau du PE d'entrée au réseau, identifie le chemin via lequel il doit envoyer le trafic et c'est le protocole PSC qui permet la synchronisation entre les ponts « sélecteur » (en entrée et en sortie) pour pouvoir identifier le chemin actif.
- La protection d'anneau: ce modèle est utilisé dans les scénarios point à multipoint. Plusieurs régimes ont été normalisés: le Re-routage Rapide (Fast reroute FRR), la commutation de protection multipoint, et le mécanisme G.8132-like.

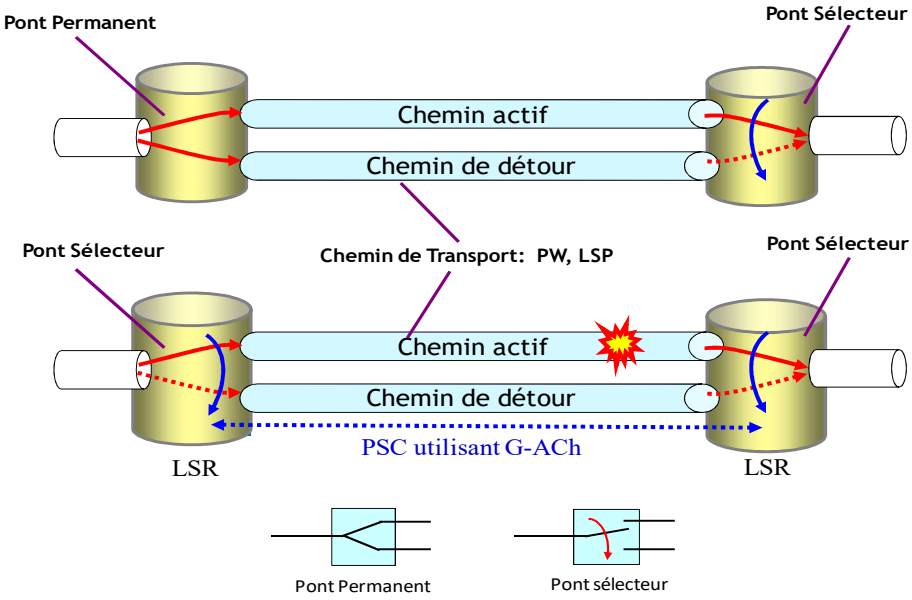


Figure 14 MPLS-TP : Mécanismes de protection et de restauration

Enfin, il est toujours possible d'avoir une protection basée sur le plan de donnée et qui hérite des mécanismes existant utilisés par GMPLS (RSVP-TE) et PW (statique ou via T-LDP).

d) Opération, administration et maintenance

La technologie MPLS-TP a profité de l'héritage des réseaux de transport classiques pour disposer d'une suite OAM très robuste et très riche. Ces OAM permettent de garantir des niveaux de services aux clients (SLA), de définir les mécanismes de protection et de restauration de chemin de transport, de localiser les pannes, de vérifier efficacement la continuité de service, d'offrir des capacités de contrôle de la qualité et de mettre à la disposition du fournisseur un réseau multiservice.

Le chapitre suivant traite en détail les outils OAM et les différences entre les deux standards et met en évidence leur problèmes d'interopérabilité et d'Interfonctionnement.

2.4. Conclusion

Les réseaux de transport Ethernet sont les fruits d'évolution et de fusion de plusieurs technologies. PBB-TE et MPLS-TP sont des technologies de transport de classe « carrier Ethernet » offrant plus ou moins les mêmes fonctionnalités. Au niveau du plan de gestion, les deux technologies utilisent le concept d'approvisionnement (ou configuration) et de surveillance statique en se basant sur un système de gestion et de supervision NMS et supportent, en option, l'usage d'un plan de contrôle dynamique.

Il existe toutefois des services que la technologie MPLS-TP peut offrir comme les services de types E-LAN et E-TREE. La technologie PBB-TE se limite au seul service E-LINE.

La pénétration faible de la technologie PBB-TE dans l'industrie n'a pas favorisé l'amélioration de cette norme IEEE. La plupart des fournisseurs et opérateurs, y compris certains des premiers bailleurs de fonds de PBB-TE comme Ciena et British Telecom, ont choisi l'adoption de la technologie MPLS-TP. Cependant, le procédé d'encapsulation PBB utilisé dans les réseaux PBB-TE est encore utilisé dans de nombreux réseaux et également proposé dans plusieurs nouveaux projets de redimensionnement de réseaux Ethernet notamment celui des centres de données.

Dans la suite de notre travail, nus nous focaliserons uniquement sur la technologie MPLS-TP. Cette dernière possède des caractéristiques qui découlent de technologies diverses dont les normes ont été développées par diverses institutions comme IETF ou ITU-T. Dans le prochain chapitre, nous discuterons des caractéristiques des outils OAM utilisés par les réseaux de transport Ethernet, nous exposerons également les problématiques liées à l'utilisation de différentes normes, et nous étudierons les solutions proposées afin de résoudre ce genre de problèmes.

Chapitre 3

Les outils d'opération, d'Administration et de Maintenance

3.1. Introduction

Les outils OAM jouent un rôle important dans les réseaux de transport, puisqu'ils fournissent les moyens de gestion de pannes et les mécanismes de suivi de performance au niveau transport et service, et permettent de délivrer les services, sous contrainte de respecter les accords d'engagement sur le service (SLA), tout en réduisant les coûts opérationnels.

Ces OAM offrent un ensemble complet de fonctionnalités qui opèrent sur le plan de données fournissant pour certains des mécanismes orientés réseau utilisés pour surveiller l'infrastructure du réseau afin d'améliorer le comportement général du réseau et le niveau de performance, et pour d'autres des mécanismes axés sur les services utilisés pour surveiller les services offerts aux clients finaux. Ces mécanismes permettent de réagir rapidement suite à une panne réseau et de faciliter la vérification de certains critères liés aux SLA (comme le taux de pertes de paquets, le délai, la gigue…). Les mécanismes de gestion de défauts sont utilisés pour la détection et la localisation de pannes ainsi que pour le diagnostic et la notification. Les autres mécanismes de gestion de performance permettent le suivi de la qualité de service par rapport aux critères SLA (par exemple, la gigue, la latence et de perte de paquets).

Les sections suivantes présentent les différentes familles OAM qui représentent un intérêt par rapport aux réseaux de transport Ethernet. Nous mettons en évidence les problématiques d'interopérabilité qui peuvent découler de la présence de plusieurs types de standards. Nous présentons également un état de l'art des différentes solutions existantes permettant de résoudre ce genre de problème.

3.2. Les Outils OAM

MPLS-TP supporte trois types d'OAM: Saut-par-saut, Out-of-band OAM et in-band. Le modèle OAM in-band, qui se base sur le canal associé ACH (Associated Channel), a été adopté par MPLS-TP et même généralisé au niveau LSP. Ainsi, les paquets OAM peuvent partager le même chemin qu'emprunte le trafic utilisateur, et fonctionner par domaine ou à travers de multiples domaines. Ce modèle n'est pas lié au plan de contrôle.

Le canal associé (ACH) est connu, historiquement, comme technique in-band permettant la vérification de la connectivité d'un Circuit VCCV (Virtual Circuit Connection Verification) et est applicable uniquement aux Pseudowires. Par contre, au niveau LSP, il n'existe pas de moyens pour différencier les paquets utilisateur des paquets OAM.

Dans MPLS-TP, le concept du PW ACH a été étendu pour devenir le «Generic Associated Channel» (G-ACH) comme indiqué sur la Figure 15. Il y a eu également l'introduction d'un nouveau Label, G-ACh Alert Label (GAL), qui permet d'identifier les paquets G-Ach[42]. L'objectif étant de disposer d'une boîte à outils OAM de bout-en-bout qui permet aux transporteurs d'utiliser les OAM à chaque niveau du réseau: LSP, pseudowire et section.

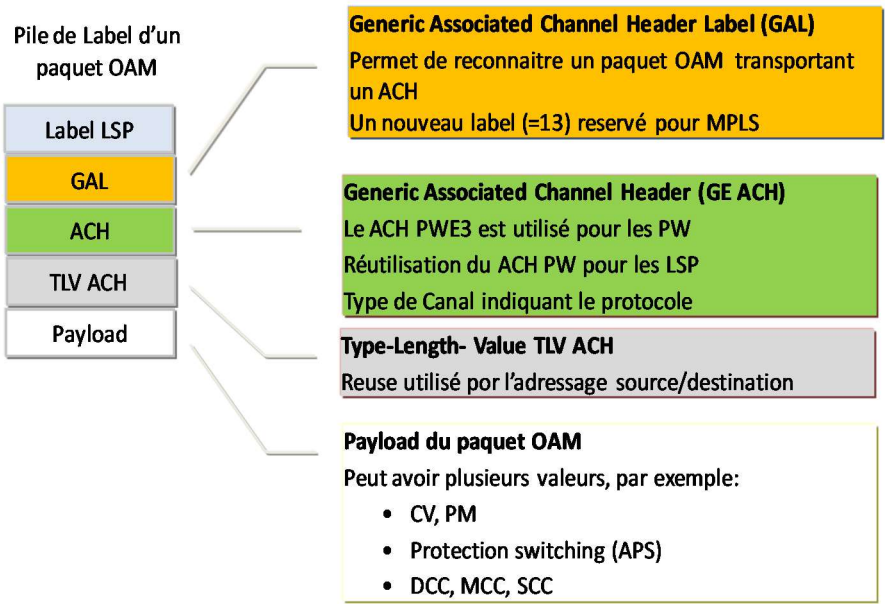


Figure 15 Format d'un paquet OAM

Nous pouvons diviser les outils OAM en deux grandes classes [43] comme indiqué dans le Tableau 3:

- Continu (proactive) avec 3 niveau de surveillance: Statut (vérification de la continuité et de la connectivité), Performance (perte de trame) et maintenance (suppression d'alarmes, indication de verrouillage, indication d'échec à distance et indication du signal client).
- A la demande (réactive) avec 3 niveaux de surveillance: statut (de vérification de la connectivité), performance (perte de trame, délai, gigue et débit) et isolement de défaillance (connectivité d'un chemin et la connectivité d'un flux).

Fonctions OAM	OAM MPLS-TP (IETF)	
	Continu (proactive)	À la demande (réactive)
Vérification de la continuité	BFD	Vérification de la continuité
Vérification de la connectivité (vérification du chemin)	BFD	Vérification de la connectivité (vérification du chemin)
Gestion des performances	LM et DM	Gestion des performances
Localisation des défauts	LDI	Localisation des défauts
Intégrité du site distant	BFD	Intégrité du site distant
Signal d'alarme	AIS/RDI	Signal d'alarme

Tableau 3 Les deux grandes classes d'OAM (IETF)

Où : LM: Mesure de perte; DM: Mesure de délai; FM: Gestion des défauts

Les deux organismes de standardisation ITU-T et IETF ont chacun proposé sa propre vision d'OAM MPLS-TP. Tandis que ITU-T pousse à utiliser la norme G.8113.1 basée sur les Ethernets OAM Y.1731, IETF propose la norme G.8113.1 sous forme d'extensions de mécanismes BFD et LSP ping/traceroute basé sur les OAM MPLS.

Dans les sections suivantes, On présente les différents types d'OAM ainsi que leur domaine d'utilisation et mode de fonctionnement.

3.2.1. Les OAM Ethernet : définis par ITU-T

Les mécanismes OAM Ethernet constituent la capacité de permettre à un fournisseur de services de créer, surveiller et dépanner des connexions et des services Ethernet de manière standardisée. Elles aident les fournisseurs de services à offrir une assurance de service de bout en bout allant du site client et passant par le cœur IP/MPLS et/ou par un réseau Metro Ethernet. Les protocoles suivants constituent les fondements des outils Ethernet OAM, Figure 16:

- IEEE 802.1ag: Connectivity Fault Management (CFM) [44].
- La recommandation UIT-T Y.1731: liée aux fonctions et mécanismes OAM des réseaux Ethernet. En fait, la recommandation UIT-T Y.1731 reprend en partie les spécifications de la norme IEEE 802.1ag en ce qui concerne les fonctions de surveillance et propose de nouvelles fonctions liées, principalement, à la performance [45].
- IEEE 802.3ah: OAM Ethernet du Premier Mile (First Mile) [46].
- MEF E-LMI: Local Management Interface Ethernet [47].

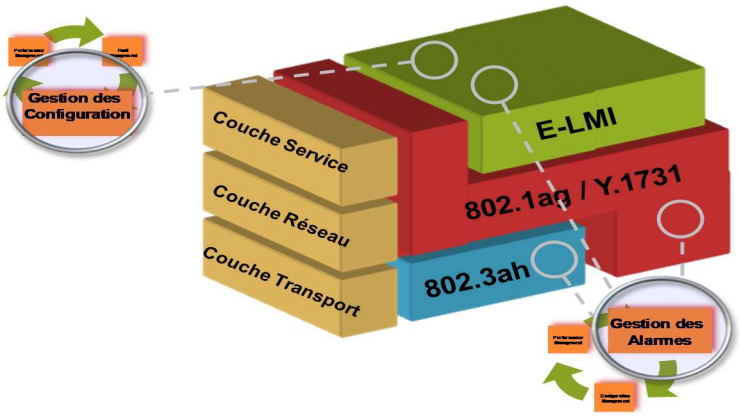


Figure 16 Les OAM Ethernet

a) Gestion des défauts de connectivité: CFM

Les outils CFM (ou outils de Gestion des défauts de connectivité) permettent de surveiller l'état des liens réseau de bout en bout d'un service de transport, comme un VLAN ou un circuit EoMPLS (Ethernet over MPLS), qui se prolonge à travers deux ou plusieurs dispositifs réseau [44]. Les CFM surveillent directement l'état des services, et donc, surveillent indirectement aussi le support physique transportant ces services et peuvent aider à identifier les problèmes de connectivité de réseau résultant de mauvaise configuration ou de problèmes physiques. On peut faire appel aux CFM pour surveiller un seul point du réseau, une liaison point-à-point, et des services multipoint gérés par un ou plusieurs fournisseurs de services.

L'extrémité de maintenance MEP (Maintenance End Point) est créée en point d'entrée ou de sortie de livraison du service à surveiller. Une MEP, située au point de départ d'un service sous surveillance, génère des messages qui sont reçus par une autre MEP située au point de terminaison de ce service.

Les points intermédiaires de maintenance MIP (Maintenance Intermediate Point) sont créés sur les ports des équipements transportant le service entre les MEP afin de traquer les défauts aux niveaux des points intermédiaires comme indiqué sur la Figure 17. Un groupe de MEP, opérant pour surveiller un service donné, définissent une association de maintenance (MA).

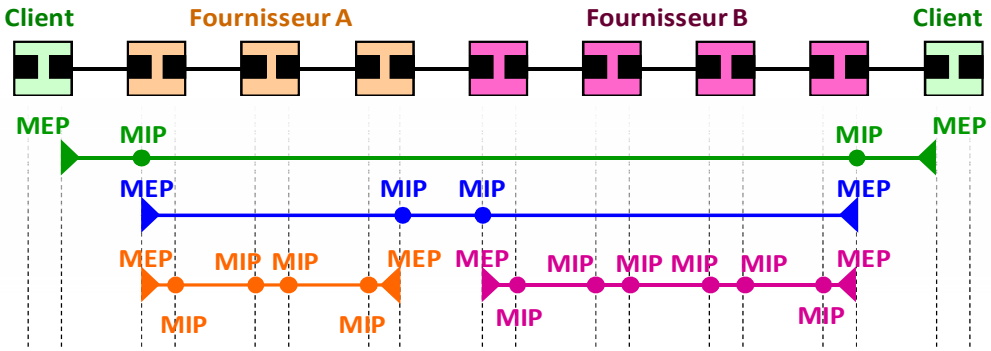


Figure 17 CFM : Concept des Points de Maintenance

Il existe plusieurs types de fonctions CFM responsable chacune d'un type de surveillance. Chaque fonction utilise son propre format de messages CFM et par conséquent il est courant d'utiliser des noms de message de CFM comme des synonymes pour les fonctions CFM respectives.

1. Support Multi-Domaine

Les CFM supportent la surveillance multi-domaine, qui est une fonction très utile surtout quand un service Ethernet entre sites distants nécessite des connexions à travers des équipements installés et contrôlés par plusieurs entités. Chacune de ces entités est responsable de surveiller l'état des services qu'ils fournissent afin d'être informée de toute perte ou perturbation de service à tout moment.

Afin de surveiller le même service à travers les différents équipements gérés par les différentes entités du fournisseur, un MD (Domaine de Maintenance) est attribué pour chaque entité et une MA (Maintenance Association) est créée pour former la structure dans laquelle les différentes entités de gestion seront contenues. On peut atteindre jusqu'à huit MD pour un seul service, ce qui permet d'identifier la position par rapport au domaine CFM adjacent. Les valeurs MD affectées doivent être soigneusement choisies, en affectant la valeur numérique la plus faible utilisée aux niveaux les plus « internes » du domaine et la valeur numérique la plus haute dans l'ordre où ils sont configurés vers l'extérieur des extrémités du service. Les MD doivent être entièrement imbriqués, ainsi un domaine intérieur doit être entièrement contenu dans un domaine environnant.

Le niveau d'un MD, qui est par défaut égale à 3, ne doit pas nécessairement être modifié pour configurer un CFM. Ce niveau ne pourrait être changé que lorsque le fournisseur du CFM a besoin de fournir un niveau MD supérieur à 3. La Figure 18 montre des exemples de cas avec différentes combinaisons de valeurs MD et MA

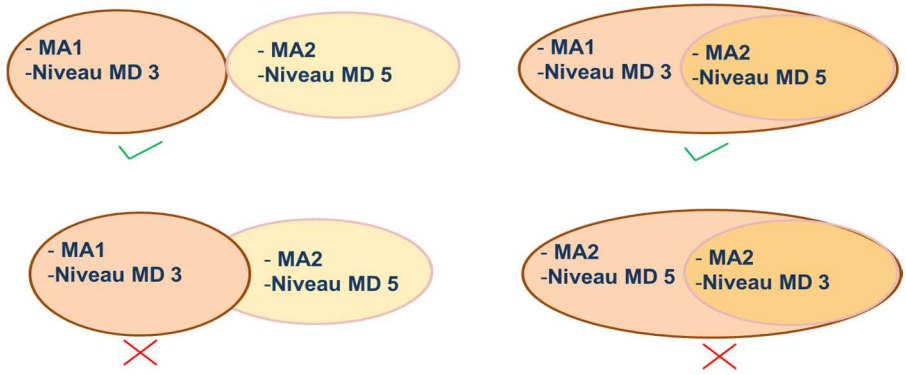


Figure 18 CFM : Multi-Domain vs Association de Maintenance MA

2. Messages de contrôle de continuité : Continuity Check Messages (CCM)

Les messages de contrôle de continuité (CCM) sont envoyés le long d'un VLAN ou un autre service pour déterminer l'état de la connexion du service. Ceux-ci peuvent être considérés

comme des messages de « heartbeat » (battement de cœur). Les messages CCM sont vus comme des messages multicast, par les équipements non configurés en CFM, et vont alors être transférés aux autres équipements. Si une MEP, sur un périphérique réseau, ne reçoit pas un CCM dans le délai prévu, un événement de type service en défaut est alors généré.

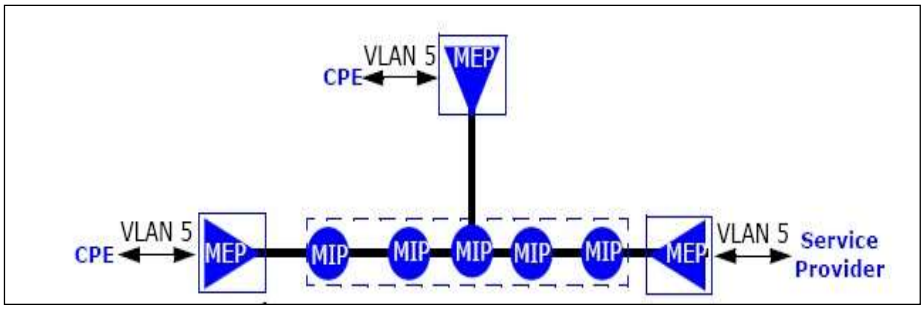


Figure 19 supervision d'un VLAN à l'aide de CFM

Dans la Figure 19 le service surveillé est VLAN 5. La MA est constitué de trois MEP (un sur chaque bord du service) et de plusieurs MIP. L'un des MEP surveille le VLAN 5 à sa source, ce qui correspond à la connexion au fournisseur de service. Les deux autres MEP surveillent VLAN 5 au niveau des équipements installés chez le client (CPE). Ainsi, pour pouvoir surveiller les services multipoint-à-multipoint, des CCM sont envoyés avec une adresse de multicast Ethernet alors que dans le cas des services point à point des adresses unicast et multicast peuvent être utilisées.

Outre la surveillance continue d'un service, les CFM supportent également d'autres fonctions de dépannage moyennant des opérations de bouclage et de trace des liens.

3. Messages Loopback (MLB)

Pour pouvoir vérifier la connectivité vers d'autres MEP ou MIP pour une MA spécifique, la MEP envoie des messages de bouclage MLB (Message Loopback). La fonction Loopback ressemble, dans son fonctionnement, à la méthode « ping » avec un mode requête / réponse. Une requête Loopback nécessite une réponse de type RLB (Réponse de Message Loopback), Figure 20, ainsi les message MLB et RLB sont utilisé pour vérifier l'état des connexions bidirectionnelle et sont généralement initiées par une commande de l'opérateur. Toutefois, une MEP peut être programmé à envoyer des MLB d'une manière périodique. Il est à noter que les Loopback sont envoyé en mode unicast pour la norme IEEE 802.1ag, et en unicast et multicast pour la norme Y.1731.

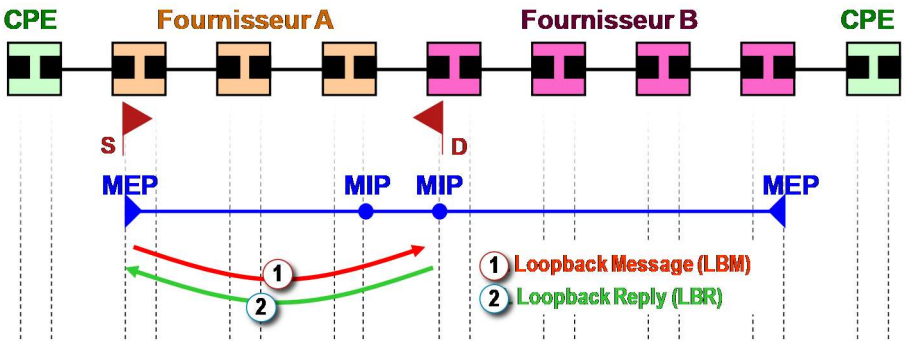


Figure 20 Messages Loopback

4. Message de trace de lien (MTL)

Les MEP envoient des messages de « Trace de lien » MTL (Message de Trace de Lien) pour un MA donné afin d'identifier les relations d'adjacence avec d'autres MEP ou MIP du même niveau de maintenance MD, ou aussi pour aider à isoler un défaut sur la chaine bout-en-bout. Les messages MTL contiennent des informations comme l'adresse MAC de la MEP cible qui termine le trace du lien. Quand une MIP ou MEP reçoit un message MTL, il génère une réponse unicast RTL (Réponse de Trace de Lien) et l'envoi au MEP initiateur du trace et transmet aussi le MTL à l'adresse MAC destination de la MEP cible, Figure 21. Un MTL permet de retracer efficacement le chemin vers la MEP cible.

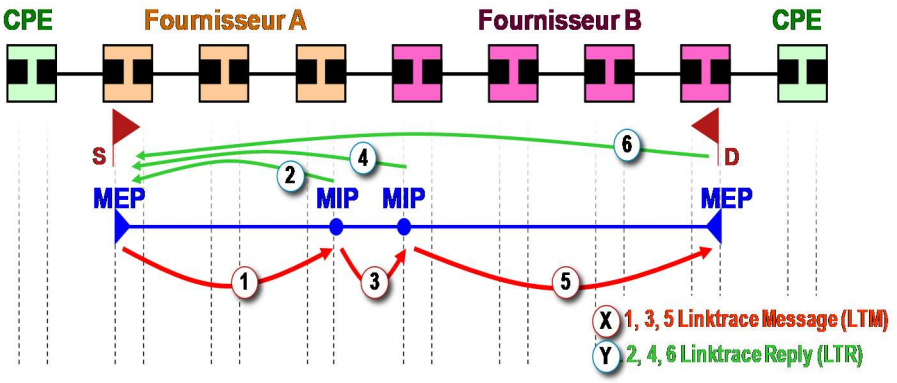


Figure 21 Messages Linktrace

Il existe d'autres types de messages CFM, qui sont utilisés pour fournir des informations complémentaires lorsqu'un défaut est détecté.

b) Y.1731 Surveillance des Performances

La norme Y.1731, introduite par l'UIT-T[45], complète la norme IEEE 802.1ag en proposant des fonctions de mesure de performance comme par exemple: le taux de perte de trame, la mesure de délai de trame et la mesure de débit. Y.1731 définit aussi les mêmes fonctions CFM de la norme IEEE 802.1ag (telles que CCM, MLB et MTL) avec de légères modifications comme l'utilisation d'adresses multicast contre des adresses unicast, etc. Le Tableau 4 suivant montre la terminologie utilisé par les deux normes Y.1731 et 802.1ag :

IEEE 802.1ag		ITU-T Y.1731	
ME	Maintenance Entity	ME	Maintenance Entity
MA	Maintenance Association	MEG	ME Group
MAID	MA Identifier	MEGID	MEG Identifier
MD	Maintenance Domain	---	Pas d'équivalent disponible
MD Level	MD Level	MEG Level	MEG Level
MEP	MA End Point	MEP	MEG End Point
MIP	MD Intermediate Point	MIP	MEG Intermediate Point
---	Pas d'équivalent disponible	Server MEP	Serveur MEP

Tableau 4 Terminologie des normes Y.1731 et 802.1ag

1. Taux de pertes de Trames (TPT)

La perte de trame est calculée en envoyant des compteurs dans les messages de mesure de perte MMP et dans les Réponses de mesure perte RMP. Les compteurs des deux extrémités sont alors comparés pour pouvoir calculer le taux de pertes de trames.

Les trames OAM CCM sont utilisées pour mesurer les pertes sur des scénarios multi-point. Le TPT est calculée à l'aide des paires de trames consécutives, ce qui compense l'absence de synchronisation entre les valeurs de comptage initiales.

2. Mesure de délai de trame (FDM)

Les messages de mesure de délai (DMM) et de réponse de mesure de délai (DMR) comprennent des temporisateurs qui sont utilisés pour calculer le délai de trame (FDM).

Le FDM peut être calculé pour un trajet « sens unique» ou pour un trajet « aller-retour ». Il est par contre nécessaire, pour le FDM «sens unique», d'avoir les deux extrémités totalement synchronisées à une même source pour s'assurer qu'il n'y a pas de décalage au niveau de l'horodatage et ainsi avoir une mesure précise.

Le FDM « aller-retour » ne subit pas cette contrainte de synchronisation sauf s'il existe une imprécision de traitement DMM-DMR au niveau de la MEP cible.

3. Débit

Les recommandations Y.1731 pour le calcul du débit sont basées sur le RFC 2544 [48], qui préconise la mesure de débit en envoyant des trames à un débit incrémentale (jusqu'au débit théorique maximal), tout en reportant le pourcentage des trames reçu ainsi que le

débit à partir duquel les trames sont supprimées ce qui est en général dépendant de la taille des trames envoyées.

Certains mécanismes spécifiés au niveau de la norme Y.1731, comme le bouclage inséré par une MEP, peuvent servir à simplifier les scenarios de mesure de débit.

c) IEEE 802.3ah: OAM de lien Ethernet

La norme IEEE 802.3ah [49] , appelée également Ethernet in the First Mile (EFM), comporte trois types de supports physiques et de topologies:

- point-à-point: en paires torsadées à 10 Mbit/s sur une distance de 750 m ;
- point-à-point: en fibre optique à 1 Gbit/s sur une distance de 10 km ;
- point-à-multipoint: en fibre optique à 1 Gbit/s sur une distance de 10 km.

Elle permet de définir une sous-couche OAM de liaison de données au sein de la couche 2 du modèle OSI, Figure 22, en prévoyant des mécanismes pour le fonctionnement de la liaison de surveillance tels que « l'indication de défaut à distance » et « le contrôle de bouclage distant ». Il est important de noter que cette fonctionnalité OAM s'applique seulement aux liaisons Ethernet directement connectées reliant les ports Ethernet voisins, et il est de la responsabilité des protocoles des couches supérieures comme le CFM de mettre en œuvre des fonctions OAM de bout-en-bout (sur plusieurs liens) dans un réseau.

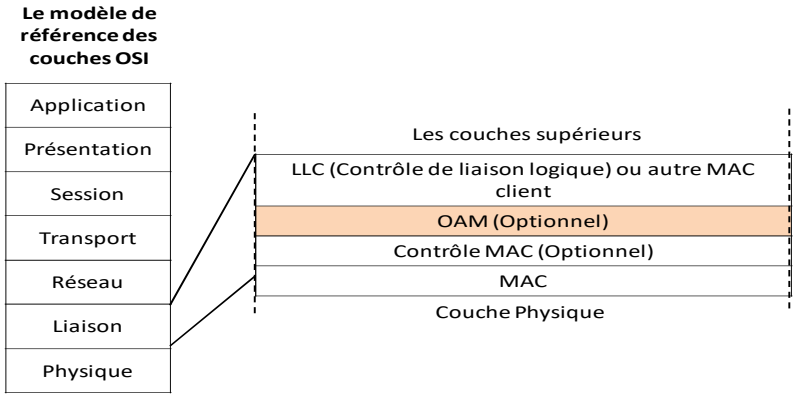


Figure 22 Positionnement des OAM EFM au niveau de la couche 2 du modèle OSI

L'une des fonctions clés, qui peuvent être mis en œuvre à l'aide des OAM des liens Ethernet, est le bouclage à distance qui est un mécanisme utilisé par un équipement de terminaison de donnée (DTE) par lequel il demande au DTE distant de passer en mode de bouclage. Dans ce mode, toutes les trames envoyée sont, tout simplement, renvoyées inchangées au DTE source. Les trames de retour peuvent ensuite être analysées par l'expéditeur afin de déterminer la qualité de la liaison.

D'autres types de messages EFM existent comme par exemple : PDU d'information, notification, demande/réponse d'une variable…

d) Metro Ethernet Forum Ethernet Local Management Interface (E-LMI)

Le Metro Ethernet Forum Ethernet Local Management Interface (E-LMI) offre aux équipements client la possibilité de recevoir les informations concernant l'état et les attributs des services Ethernet qui permettent de garantir une configuration automatique et une amélioration de la performance du réseau d'accès client [50].

Le protocole ELMI est basé sur la norme UIT-T Q.933, X.36, et sur le Frame Relay Local Management Interface (FR-LMI).

En utilisant les messages E-LMI, l'équipement client peut demander et recevoir, par exemple, des informations relatives à l'état de son circuit virtuel qui peut être "nouveau", "actif" ou "inactif", ou aussi des informations lié à la configuration telles que le correspondance ID CE-VLAN / ID de service Ethernet ou aussi son profil de la bande passante.

3.2.2. Les OAM MPLS-TP: définis par IETF

Le protocole MPLS dispose déjà de quelques outils OAM comme: le ping LSP, le traceroute LSP, le BFD et le mécanisme VCCV. Ces mêmes mécanismes OAM ont été pris comme base, par l'organisme IETF, pour développer et étendre les OAM MPLS-TP. La Figure 23 résume les différentes méthodes OAM, définies par IETF, supportées par MPLS-TP au niveau LSP et au niveau PW. Nous exposerons en détail chacun de ces outils OAM dans les sections suivantes.

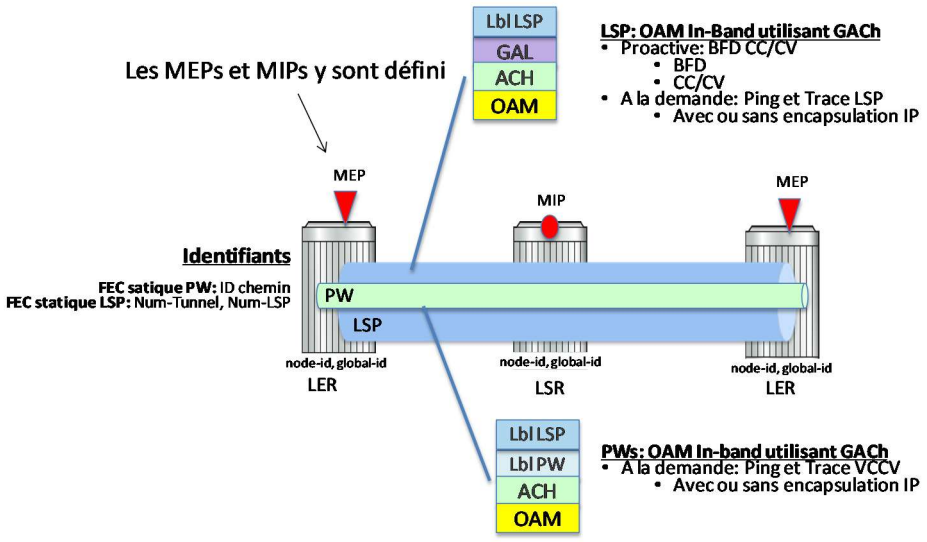


Figure 23 Les OAM MPLS-TP définis par IETF

a) Ping et Traceroute LSP

Le Ping et le Traceroute LSP sont des implémentations qui existent déjà au niveau du standard MPLS et qui elles aussi sont basées sur le même concept du Ping ICMP. Le LSP

Ping fournit un mécanisme pour détecter les défaillances d'un LSP au niveau du plan de données. Le Ping et le trace LSP se basent sur le modèle Echo/réponse pour pouvoir détecter et localiser des défauts dans les réseaux de transport [51].

En effet, L'OAM LSP ping effectue des tests de connectivité à l'intérieur du LSP (in-band). Le routeur LER Ingress (Label Edge Router d'entrée au réseau), à l'origine du Ping LSP, crée un paquet MPLS de demande d'écho (request) pour le LSP et le chemin MPLS à tester. Après avoir envoyé le paquet de demande d'écho MPLS, une réponse d'écho MPLS du LER Egress (LER de sortie) qui termine LSP est générée. L'état du LSP est ainsi déterminé lorsque le paquet de réponse d'écho est reçu par le LER Ingress, Figure 24.

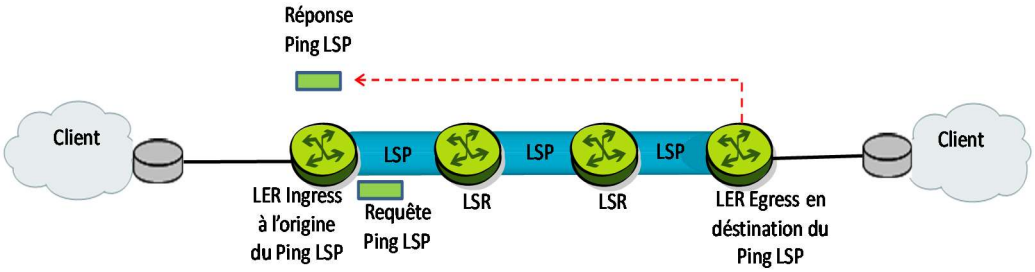


Figure 24 Ping LSP

Concernant le Traceroute LSP, le LER Ingress envoi une requête de demande d'écho pour le LSP à tester. Ce paquet contient une valeur Time to live TTL, valeur qui ne dépasse pas 255, qui va être incrémentée au fur et à mesure qu'il traverse les différents routeurs MPLS (LSR : Label Switched Router). Ces LSR envoient une réponse au LER Ingress à la réception du paquet écho et incrémente la valeur du TTL. Cette opération continue jusqu'à dépassement du TTL ou jusqu'à l'arrivé du paquet écho au LER Egress.

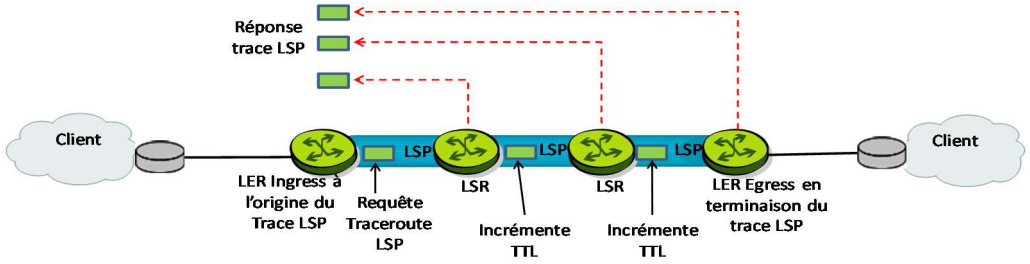


Figure 25 Traceroute LSP

b) Bidirectional Forwarding Detection

Le Bidirectional Forwarding Detection BFD est un protocole qui a été défini par l'IETF et qui a comme objectif la détection rapide de pannes entre deux équipements adjacents afin de pouvoir déclencher par la suite les mécanismes de protection adéquat [52]. Le protocole BFD fonctionne exclusivement en mode point-à-point bidirectionnel et offre des temps de détection de l'ordre de millisecondes. Ce protocole peut être implémenté au niveau matériel grâce à la taille fixe de ses paquets. Ainsi, il existe plusieurs mécanismes de protection ou de redondance de lien ou de nœud de réseau qui se déclenchent en se basant sur les changements d'état du BFD. Le protocole MPLS-TP utilise les sessions BFD pour vérifier

la continuité à la fois au niveau du tunnel MPLS (LSP) ou au niveau du tunnel de service (PW).

Le protocole BFD peut opérer en deux modes: Synchrone (à la demande) et Asynchrone (proactif). Il existe deux modes de fonctionnement pour les LSP bidirectionnels: celui dans lequel l'état de session des deux directions du LSP est coordonné, c'est donc une seule session bidirectionnel BFD qui est utilisée; et celui construit à partir de sessions BFD de manière à ce que les deux directions fonctionnent indépendamment mais appartenant toujours au même MEG. On aura donc deux sessions BFD indépendantes qui sont utilisées pour ce type de fonctionnement. Une session BFD est identifiée au niveau de chaque extrémité par des discriminateurs locales qui sont échangés au moment de l'établissement de la session. La fréquence d'envoi de message BFD peut être négociée entre les deux extrémités.

Lors d'une session bidirectionnelle, chaque nœud envoi les messages de contrôle contenant l'état actuel du LSP ou du chemin, ainsi que les intervalles de transmission et de réception : T_{tx} et T_{rx}. Quand un nœud reçoit un message, il ajuste son intervalle de réception avec l'intervalle de transmission demandé par l'autre extrémité si $T_{tx} \geq T_{rx}$. L'état du LSP ou du chemin est déclaré « hors service » (down) si un nombre identifié de paquets BFD n'est pas reçu pendant une durée déterminée T_{det}. La Figure 26 présente la machine d'état concernant le protocole BFD pour un seul nœud montrant les différents états possible: Down, Init et Up.

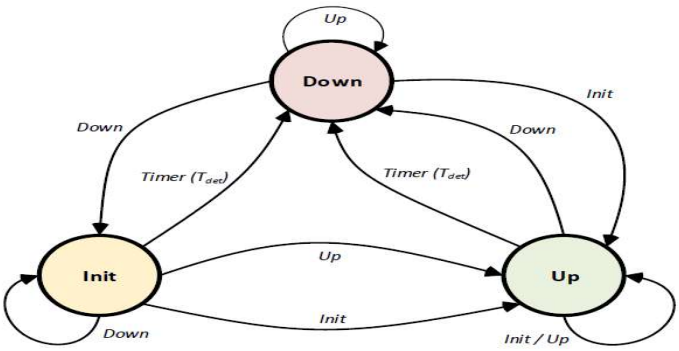


Figure 26 BFD : Machine d'état

c) Virtual Circuit Connectivity Verification VCCV

Les outils OAM VCCV fournissent les mécanismes de détection de défaut et de diagnostic des tunnels de service PW en envoyant leurs paquets in-band avec le trafic utilisateur. La supervision des tunnels de service se fonde typiquement sur l'utilisation du canal VCCV pour tester la connectivité et détecter un défaut sur un PW. Ce même canal peut également notifier au LER distant un défaut au niveau du circuit d'attachement local (AC). Ce canal permet d'utiliser les protocoles tels que BFD ou LSP Ping au niveau du tunnel de service.

Le protocole VCCV supporte plusieurs types de mécanismes OAM : ICMP Ping, LSP Ping, et BFD. Il se base sur deux composantes :

- Une composante de signalisation qui permet de communiquer les capacités supportées du VCCV,
- Une composante de commutation qui permet de traiter la charge utile (payload) du PW comme un paquet de contrôle.

3.3. Etat de l'art sur les solutions d'interopérabilité OAM existantes

Dans cette partie nous présentons les problématiques d'interopérabilité liées au fait d'avoir plusieurs standards dans un réseau de transport MPLS-TP.

Nous étudierons également les travaux réalisés pour répondre aux problématiques de l'interfonctionnement des outils OAM et aussi de leur utilisation par le plan de contrôle.

3.3.1. La problématique des outils OAM dans les réseaux MPLS-TP

Les deux normes OAM citées auparavant, celle de l'ITU-T et celle de l'IETF, prétendent satisfaire aux exigences du réseau de transport MPLS-TP. Cependant, il existe de grandes différences entre ces deux normes notamment en ce qui concerne le format de leurs messages Packet Data Unit (PDU), l'exemple de la Figure 27 ci-dessous illustre bien cette différence au niveau de la structure des trames PDU. Cette différence dans le formatage des paquets rend la communication, d'un point de vue domaine de gestion, impossible. A cause de cette interopérabilité, un service qui est déployé sur un réseau MPLS-TP, dont un ou plusieurs équipements utilisant un standard OAM diffèrent du reste du réseau, ne pourra pas bénéficié des fonctionnalités des outils OAM.

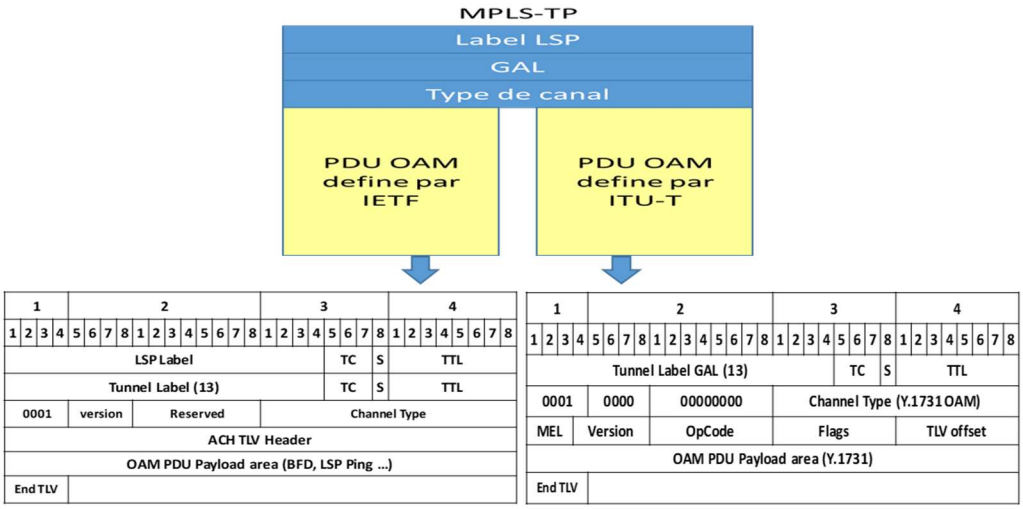


Figure 27 Format des trames PDU OAM: IETF vs ITU-T

On se trouve alors face à un problème de taille qui est celui de ne pas pouvoir monitorer un service traversant plusieurs domaines MPLS-TP et ou IP/MPLS. En effet, les opérateurs de transport Ethernet ont l'obligation de fournir des outils OAM cohérents au niveau des différentes couches/technologies (Couche 2, PW, LSP) et sur différents niveaux de

démarcation (chemin, segment, segments multiples) de façon transparente pour toutes les composantes du service vendu. Toute solution proposée doit prendre en compte le fait que les outils OAM doivent être assurés d'une manière continue et pour un service de bout-en-bout.

Hormis ce problème d'interopérabilité des OAM dans les réseaux de transport Ethernet, il est courant chez les opérateurs de fournir des SLA à leurs clients et qu'ils utilisent les outils OAM pour s'assurer que ces SLA sont respectés. Dans le cas où les services clients transportés empruntent des chemins ne respectant pas ces critères SLA, les opérateurs reprogramment généralement ces services pour qu'ils soient re-routés sur d'autres chemins. Dans le chapitre suivant, nous allons également adresser ce genre de problématique et proposer d'élargir le domaine d'application de ces outils puissants. En effet, il a été dit précédemment que les outils OAM permettaient de détecter, d'identifier et de localiser les défauts au niveau du plan data d'un réseau de transport. Il n y a donc pas de lien direct entre le plan de gestion et le plan de contrôle. D'autant plus que dans la majorité des implémentations actuelles des réseaux de transports, la partie « plan de contrôle » est souvent définie d'une manière statique (manuellement provisionné par l'administrateur du réseau via son NMS). Notre travail se focus alors sur l'utilisation du « plan de gestion » pour alimenter les décisions au niveau du « plan de contrôle » tout en éliminant les problèmes d'interfonctionnement des outils OAM.

3.3.2. Solutions existantes pour l'interfonctionnement entre les différents outils OAM

Plusieurs recherches ont été faites pour résoudre cette problématique d'interfonctionnement entre différents standards OAM. Parmi ces recherches, on trouvera deux tendances :

- La première solution consiste à rajouter une couche « fine » (Thin layer) permettant ainsi de construire un terrain d'entente et faire abstraction ainsi des incompatibilités de chacune des méthodes OAM.
- La deuxième est plus évidente puisqu'elle est basée sur la présence de certains nœuds ayant pour fonction principale la traduction, l'adaptation et le relais des messages OAM entre différentes zones de même type.

a) Couche « fine »

La couche « fine » (Thin Layer) est une couche réseau interconnectant de manière transparente des réseaux de transport multi-technologiques [53]. L'ajout de cette nouvelle couche est possible grâce à l'introduction d'un type de trame ayant un nouveau code Ethertype. Cette couche permet ainsi de cacher les détails et les spécificités des couches supérieurs et assure des connections point à point entre les différents nœuds de transport, Figure 28.

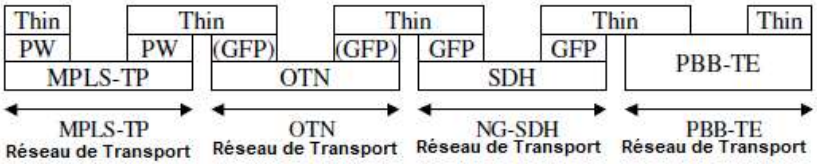


Figure 28 Illustration de la couche « fine » à travers plusieurs réseaux de Transport

Dans l'architecture à couche « fine », les mécanismes OAM doivent être effectués d'une extrémité à l'autre du service monitoré. Il est important que l'OAM de la couche « fine » interagisse avec l'OAM spécifique à la technologie de transport (MPLS-TP par exemple). Il est également important que l'OAM de la couche « fine » interagisse avec l'OAM du client, qui est responsable du monitoring de l'accès client[54].

Les mêmes principes de groupes de maintenance MEP, MIP, et niveau sont reproduits dans les OAM de la couche « fine ». A noter que chaque nœud du réseau doit être capable d'interpréter le code Ethertype afin de pouvoir traiter les messages OAM de la couche fine.

Il y a plusieurs problèmes pour réaliser pleinement l'architecture en couche « fine ». On notera notamment le coût élevé pour développer la fonction de couche « fine » sur tous les nœuds du réseau de transport. On remarque aussi que l'ajout d'une couche, peu importe sa « finesse », rendra le réseau encore plus complexe avec un nombre significatif d'entêtes, Figure 29.

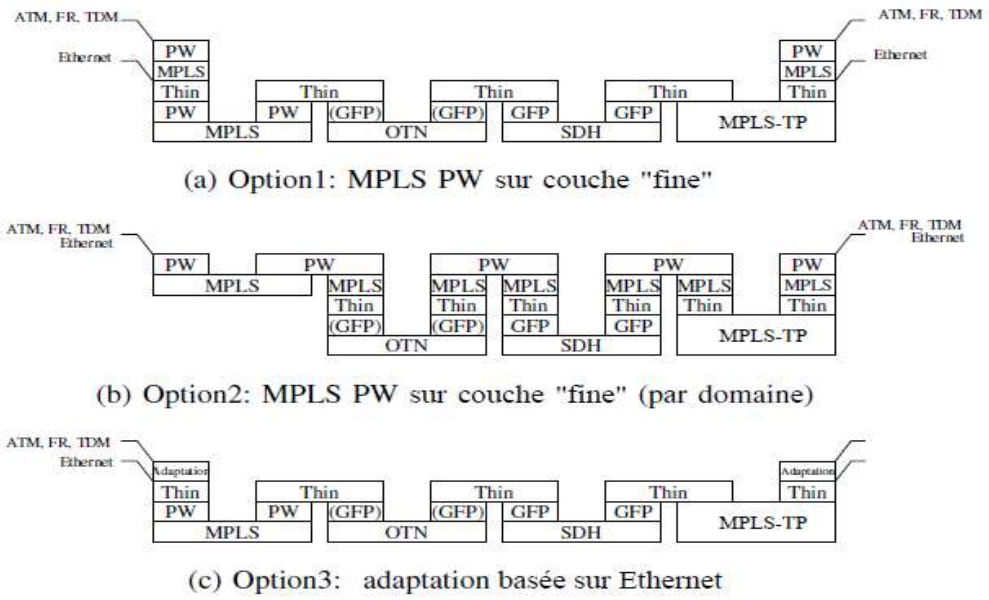


Figure 29 Illustration du rajout de la couche « fine » sur les couches d'un réseau de Transport

b) Nœud d'interfonctionnement (Interworking Node)

Le principe de la solution du « Nœud d'interfonctionnement » (Nœud IW) se base sur un concept simple qui est celui d'voir un ou plusieurs nœud qui feront office de passerelle entre plusieurs zones utilisant différentes normes OAM, [55]. Le nœud IW prendra en charge

l'interprétation, la modification et la translation de tous les messages transitant d'une zone à une autre. L'interfonctionnement est la traduction d'événements et pas nécessairement le mappage de messages à l'échelle 1:1 et peut être inter-couche et intra-couche.

Le nœud IW doit alors supporter à la fois tous les standards OAM IETF et ITU-T, mais également les standards liés aux deux plans contrôle et données. En effet, l'une des caractéristiques principales des OAM est le fait de transiter dans les tunnels de transport des données.

Dans la Figure 30, le nœud IW interconnecte deux domaines MPLS-TP ou deux standards sont adoptés, plusieurs normes sont données ici à titre d'information. Le nœud IW bénéficie d'une implémentation en doubles piles « dual stack » supportant ainsi les deux standards en même temps.

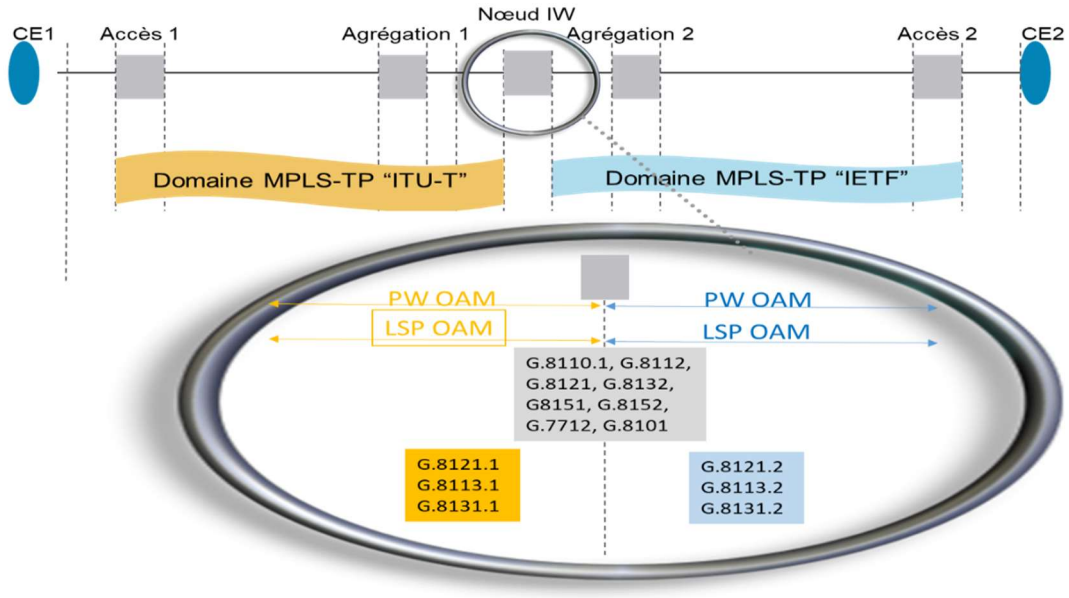


Figure 30 Schéma de principe d'un nœud IW

NB : Notons que la norme G.8113.1 est axée sur les exigences MPLS-TP OAM spécifiques basées sur le mécanisme OAM Ethernet, et que la norme G.8113.2 couvre les exigences de compatibilité OAM orientées IP/MPLS.

Le fait que le nœud IW supporte le « dual stack » implique un coût supplémentaire en matière de développement de toutes les fonctions des différentes piles protocolaires, OAM inclue. Il faudra également prendre en considération d'autres coûts comme le coût de License et de maintenance quand on pense à implémenter ce genre de solution.

A noter aussi que puisque le passage à travers le nœud IW est obligatoire, quand on livre un service d'une zone à une autre, ceci implique un risque en termes de résilience du réseau. En effet, le nœud IW représente, par nature de sa position dans le réseau, un potentiel point de défaut (Single Point of Failure SPOF).

3.3.3. Utilisation du plan de Management par le plan de contrôle

Comme cité au début de cette section 3.3, et après avoir expliqué quelles sont les alternatives actuelle pour pallier aux problèmes d'interopérabilités des OAM dans les réseaux de transport, on discutera ici des solutions proposées afin de profiter de la richesse du plan de management dans le but d'influer les décisions du plan de contrôle.

Les principales propositions dans ce créneau concernent l'utilisation d'élément de calcul de chemin appelées couramment PCE (Path Computation Element). Le PCE est une entité capable de calculer un chemin ou une route réseau à partir d'une topologie réseau. Il permet également l'application de contraintes lors du calcul du chemin. Le PCE est une application qui peut être intégrée à un nœud du réseau ou également déporté sur un serveur. Un PCE serait en mesure de calculer le chemin d'un LSP en prenant en compte certaines contraintes comme la bande passante, les coûts des liens, etc. Le RFC4655, [56], précise qu'il y a plusieurs composantes du modèle basé sur le PCE. En effet, ce modèle permet de séparer le serveur PCE du client du PCE noté également PCC. Il faudra noter également l'utilisation du protocole PCEP [57] qui est responsable des communications entre PCE et PCC ou entre différent PCE.

Le travail [58] explique comment le PCE peut tirer profit de la richesse des outils OAM pour alimenter et influencer des prises de décisions au niveau du plan de contrôle. En effet, d'après les spécifications du PCE, c'est le système de gestion NMS qui communique avec le PCE pour échanger les éléments de configurations, les contraintes SLA, et la base de donnée liée à l'ingénierie de trafic TED, Figure 31. Ainsi, les règles de forwarding (plan de données) sont communiquées directement aux PCC via le protocole PCEP, [59].

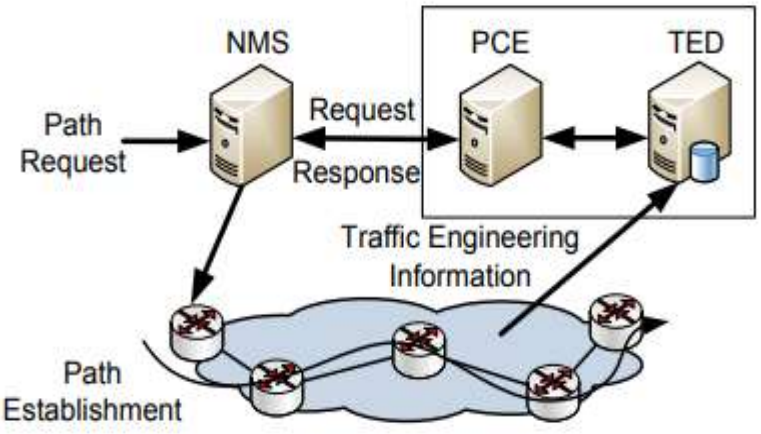


Figure 31 Architecture PCE

Dans ses échanges avec le PCE, le système de gestion NMS peut également lui communiquer les résultats des mécanismes OAM déployés dans le réseau. Avec cette opération, le « plan de Management » influe directement sur les décisions du « plan de contrôle » ce qui permet une utilisation efficace des outils OAM.

Toutefois, les PCC, qui sont aussi les LSR dans le cas d'un réseau MPLS-TP, sont contraints de fournir les informations pour alimenter la TED ce qui constitue un handicap

pour les réseaux de transport dont le plan de contrôle est statique. D'autant plus, que le PCE ne résout pas la problématique de l'interopérabilité des outils OAM puisqu'il agit uniquement sur le « plan de contrôle » et pas sur le « plan de management ».

3.4. Conclusion

Il est clair que le grand intérêt, que portent les fournisseurs et les organismes de standardisation, à la technologie MPLS-TP positionne celle-ci comme technologie de futur dans le monde des réseaux nouvelle génération de classe « carrier Ethernet ». Par contre, la diversité de méthodes et d'outils OAM peut conduire, éventuellement, à des complications liées aux problématiques d'interopérabilités avec des technologies existantes comme MPLS ou même avec des implémentations OAM d'un standard différent (IETF ou ITU-T). Dans la suite de notre travail, nous proposons également un modèle évolutif permettant d'utiliser les outils OAM pour optimiser le routage dans un réseau MPLS-TP. Ainsi on pourra mieux tirer profit de la richesse du plan de management en faveur de décisions du « plan de contrôle ».

Chapitre 4

Proposition de Modèles pour résoudre les problématiques d'interopérabilité OAM dans les réseaux MPLS-TP

4.1. Introduction

Grace à la structure de des réseaux de transport illustré par l'existence de tunnel de couches et de tunnels (tunnel de service et tunnel de transport), nous avons proposé un modèle d'architecture permettant de contourner ce problème d'interfonctionnement des OAM sans pour autant apporter des modifications dans les différentes couches protocolaires.

Deux modèles sont discutés dans ce chapitre, le premier étant le « modèle overlay » et le second étant le « modèle de cloisonnement » lequel est décliné sous deux variantes : statiques et dynamiques. Le service PW MPLS-TP est le service de base permettant de superposer les couches du réseau de transport. En effet, les services PW disposent de fonctionnalité comme le « multi-segment »-PW qui permet de concaténer les PW de chaque réseau tout en gardant une continuité de service de « bout en bout » très utilisée dans le « modèle de cloisonnement ». A la différence du « modèle overlay » qui utilise le PW comme unité de base pour transporter une couche réseau par un autre réseau dit « overlay ».

Grâce à ce concept de modèle on peut faire cohabiter deux réseaux dont chacun dispose de sa propre norme OAM. Dans la suite, on donne plus de détail sur chaque scenario ainsi que les détails concernant les déroulements des simulations ainsi que l'analyse et les commentaires des résultats de ces tests.

4.2. Proposition de Modèle Overlay

Le principe des réseaux overlay consiste à construire une topologie virtuelle à partir d'une topologie physique existante, [60], dans le but de fournir des services personnalisés à l'utilisateur. Il s'agit d'un réseau construit au-dessus d'un second réseau afin d'optimiser la distribution des données.

4.2.1. Le concept du modèle Overlay

Les réseaux overlays n'impliquent pas une modification dans l'infrastructure physique d'un réseau. Ils sont souvent utilisés comme moyen simple et peu coûteux pour déployer de nouveaux services réseaux. On citera comme exemple le réseau VPN (IPsec ou GRE) qui est un modèle de réseau overlay très répondu construit sur la base d'un réseau IP (Internet). Par analogie, le réseau qui portera le réseau overlay s'appelle le réseau Underlay, Figure 32.

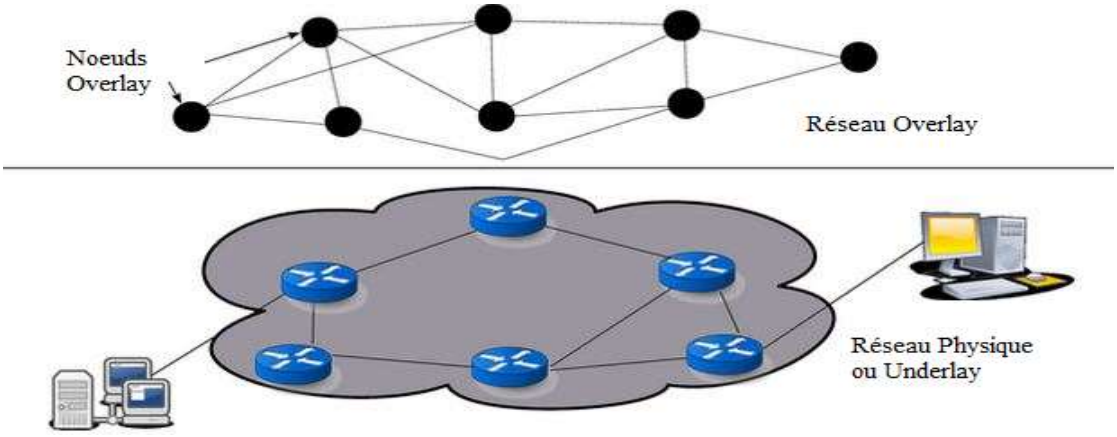


Figure 32 Principe d'un réseau overlay

Nous proposons donc d'utiliser ce concept de modèle overlay pour contourner la problématique d'interopérabilité des OAM. En effet, le fait de superposer deux couches chacune appartenant à une famille OAM permet d'éviter leur interaction.

On transportera du IP/MPLS sur du MPLS-TP ou du MPLS-TP sur du MPLS-TP sans pour autant les faire interagir. Cette situation ressemble beaucoup à une situation de Client/Serveur où la couche Overlay est cliente de la couche Underlay, Figure 33.

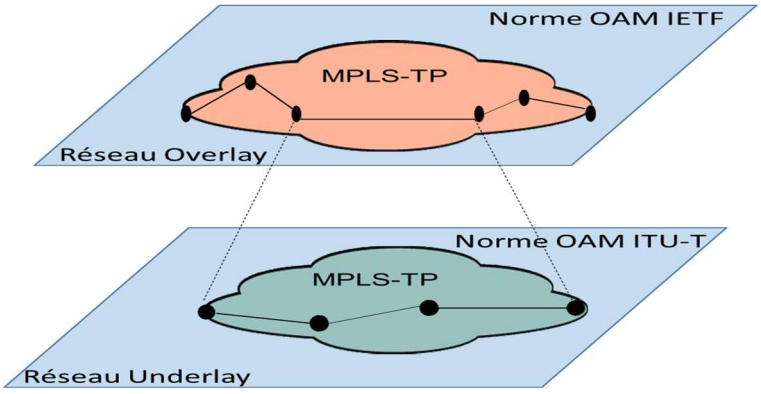


Figure 33 Modèle Overlay proposé

4.2.2. L'application du modèle Overlay aux réseaux de transport MPLS-TP

La superposition des couches est rendu possible grâce au fait qu'on peut transporter le réseau overlay en tant que service PW par le réseau Underlay. En effet dans l'architecture MPLS-TP, le PW est l'unité principale de transport des payload (charges utiles) d'autres réseaux MPLS-TP ou IP/MPLS. La Figure 34 permet d'illustrer cette démarcation nette entre les opérations MPLS-TP et les payload transportées d'un réseau (MPLS-TP) pour un autre réseau (IP/MPLS ou MPLS-TP).

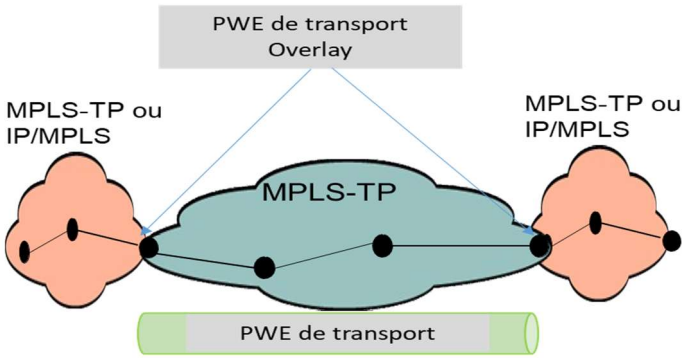


Figure 34 PW de transport : base du modèle Overlay proposé

Après avoir été correctement encapsulées au niveau d'un LER (le nœud d'entrée au réseau MPLS-TP) les données de la couche client, incluant les données du « plan de contrôle », les données du « plan de données » mais surtout celui qui nous concerne dans nos travaux qui est le « plan de gestion », elles sont transmises de manière transparente au LER qui livre le service et qui est aussi le nœud de sortie du réseau MPLS-TP. Ce canal de communication n'est autre qu'un service PW, Figure 35.

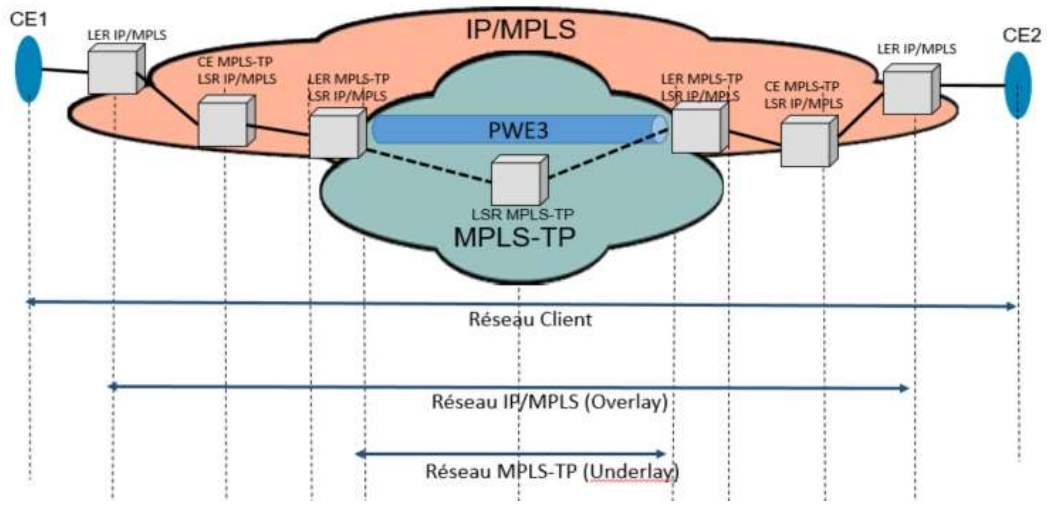


Figure 35 Modèle Overlay proposé : superposition des couches grâce aux PW

Le plan de gestion du réseau IP/MPLS est ainsi transporté d'une manière transparente en utilisant les services PW fournis par le réseau MPLS-TP. On dispose alors de deux couches d'outils OAM chacune s'exécutant dans son milieu « étanche » où aucune interaction n'est possible. Ce qui représente un contournement des problèmes liés aux interopérabilités des différentes normes OAM.

En effet, au niveau du plan de gestion, on définit un domaine de maintenance MD par type de réseau (type d'OAM), Figure 36. Les MEP procèdent alors à la découverte des « Points de Maintenance » et l'association de maintenance MA entre MIP et MEP peut alors être établie.

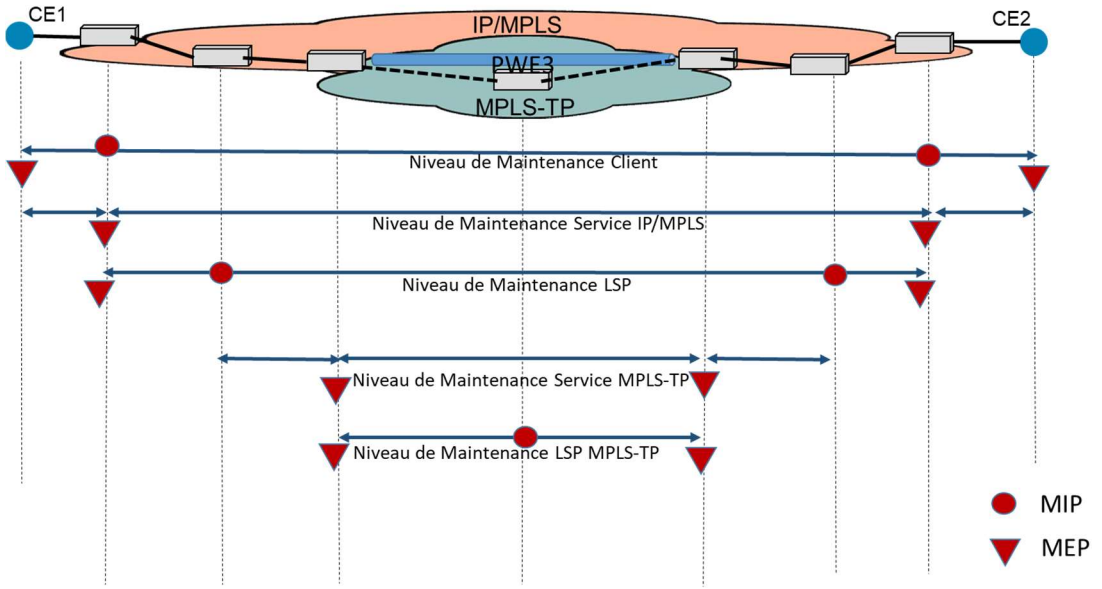


Figure 36 Différents niveaux de maintenance du modèle Overlay proposé

Plusieurs niveaux de maintenance sont alors superposés, où une dépendance est alors crée entre les différents niveaux. En effet, si par exemple un défaut (perte de connectivité) est détecté au niveau de maintenance LSP du réseau MPLS-TP, il y a automatiquement une répercussion vers le niveau de maintenance du service PW du réseau MPLS-TP. Ainsi le service PW du réseau MPLS-TP se met automatiquement en état « défaut » et une alarme est éventuellement envoyée au système de gestion NMS. L'opérateur peut ainsi traiter cette alarme et remédier au défaut.

Pour que ce défaut soit relayé au niveau de maintenance LSP du réseau IP/MPLS, le LSR IP/MPLS (qui a le rôle de CE pour le réseau MPLS-TP), doit pouvoir interpréter l'alarme AIS envoyée en aval depuis le LER MPLS-TP.

La surveillance de bout-en-bout du service client est rendue possible grâce à ce mécanisme de superpositions de couches en l'occurrence le modèle Overlay.

Le modèle overlay se caractérise donc par cette structuration OAM sur plusieurs segments: OAM physique (exemple EFM), OAM PW, et OAM LSP. Chaque segment doit effectuer ses opérations OAM indépendamment des autres segments. Les entités de maintenance, à l'intérieur de chaque domaine de maintenance, doivent utiliser le même type de standard OAM offrant ainsi les mêmes fonctionnalités et le même format PDU.

4.3. Proposition de Modèle de Cloisonnement

Le deuxième modèle proposé se base sur la segmentation des réseaux en plusieurs sous réseaux (généralement deux sous réseaux) supportant chacun une norme OAM.

L'idée est de cloisonner chaque sous réseaux de transport en définissant des frontières du domaine de maintenance. Ces frontières peuvent être représentées par un nœud de frontière ou par des liens de frontières. Chaque segment est traité à part d'un point de vue OAM, Figure 52.

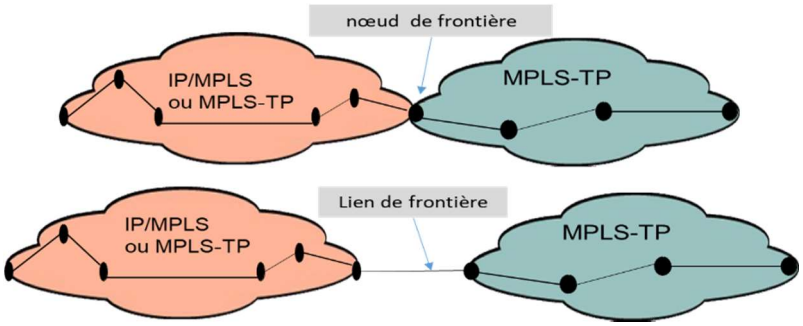


Figure 37 Cloisonnement en sous réseaux

Les LSP et les PW transportés par les réseaux MPLS-TP et IP/MPLS sont alors interconnectés au niveau des frontières. Eventuellement, toutes les contraintes du réseau MPLS-TP sont respectées à savoir: orientés connexion, pas de MultiPath (ECMP) ni de PHP.

Dans la suite, on va étudier les deux variantes proposées du modèle de cloisonnement. Le premier modèle est le « modèle statique » que nous avons pu implémenter en maquette. Par la suite, nous proposons un mécanisme d'auto découverte de MIP OAM. Ce dernier modèle se nomme le modèle de cloisonnement avec auto découverte de MIP.

Les deux sections qui suivent détaillent le fonctionnement de chacun des deux modèles proposés.

4.3.1. Concept du modèle de Cloisonnement statique

Grâce aux fonctionnalisés des pseudowires PW, [61], un service peut être transporté d'un bout à l'autre d'un réseau de transport en utilisant seul pseudowire (single PW S-PW) ou plusieurs pseudowires (Multi segment PW MS-PW).

En effet, le MS-PW est constitué de deux ou de plusieurs segments PW contigus configurés statiquement ou dynamiquement se comportant et fonctionnant comme un seul PW. Chaque MS-PW se termine sur un PE de terminaison appelé T-PE. Le nœud qui permet d'assembler deux segments d'un MS-PW s'appelle le S-PW (Switching PE). Les S-PE sont responsables de la transmission des messages OAM d'un segment à l'autre. Le MS-PW est généralement utilisé dans les réseaux hiérarchique d'une taille importante où on trouve des partie du réseau qui n'ont pas connaissance de toute la table de routage, on utilise alors le MS-PW pour prolonger un service jusqu'à ces zones-là.

Sur le plan du transport, il existe l'équivalent du principe des pseudowires multi-segment, qu'on appelle le « LSP Stitching», [62]. Le principe du « LSP Stitching» consiste à établir des tunnels LSP distincts, ces segments seront cousus ensemble pour créer un seul LSP de bout en bout.

En combinant ces deux principes, MS-PW et « LSP Stitching », on arrive à un modèle de cloisonnement permettant d'offrir des services de transport, et d'offrir également l'utilisation d'OAM de bout en bout, Figure 38.

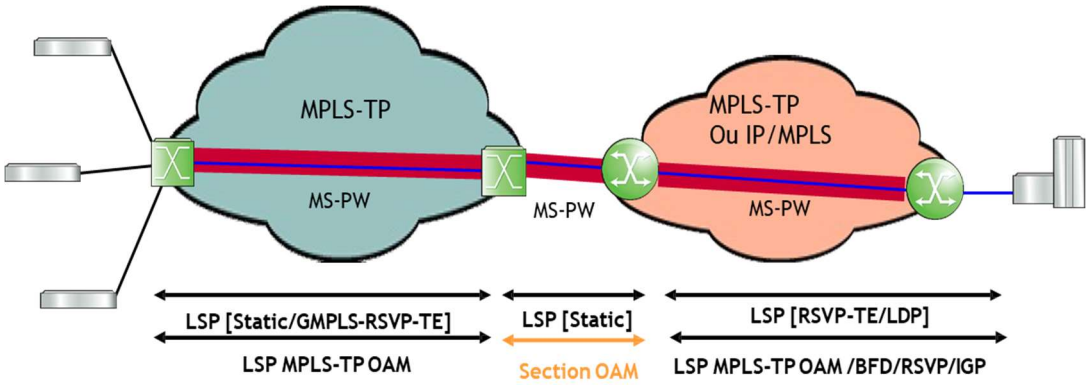


Figure 38 Principe MS-PW et « LSP Stitching »

D'un point de vue OAM et comme décrit dans la Figure 38, deux implémentations sont possible : via le « nœud de frontière » et via un « lien de frontière ».

4.3.2. Modèle de cloisonnement « nœud de frontière » :

Dans la première variante, le « nœud de frontière » joue un rôle de S-PE où il interconnecte les PW grâce aux les fonctionnalités MS-PW, et interconnecte aussi les LSP grâce au « LSP Stitching ». Le « LSP Stitching » est utilisé, à la base, pour interconnecter plusieurs aires MPLS chacune contrôlée par un protocole d'échange de label différents par exemple une aire en RSVP-TE et une autre aire en LDP. Ce modèle, que nous proposons, profite de cette fonctionnalité, ainsi que celle du MS-PW, pour interconnecter deux domaines OAM différents, Figure 39.

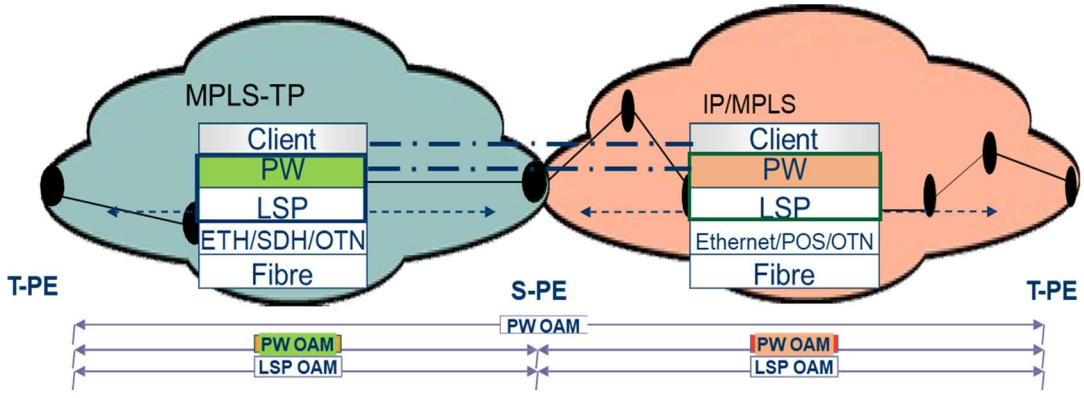


Figure 39 Modèle de cloisonnement proposé: « nœud de frontière »

Le nœud de frontière doit pouvoir supporter les deux technologies à la fois (MPLS-TP et IP /MPLS).

4.3.3. Modèle de cloisonnement « Segment de frontière » :

Autrement, la deuxième variante du modèle de cloisonnement qui est basée sur le segment de frontière, est aussi envisageable.

En effet, cette topologie suppose qu'on ne dispose pas de nœud supportant à la fois les deux familles OAM. Elle nécessite part contre l'ajout d'un segment au niveau du Pseudowire, Figure 40. Sur ce modèle seul les OAM au niveau transport (LSP) peuvent être disponible de « bout en bout », les OAM au niveau PW ne sont disponible que sur leur segment respectif. Les OAM PW ne seront pas disponible au niveau du « segment de frontière » étant donné que les nœuds S-PE ne disposent pas de fonctionnalités d'IW pour ces OAM là.

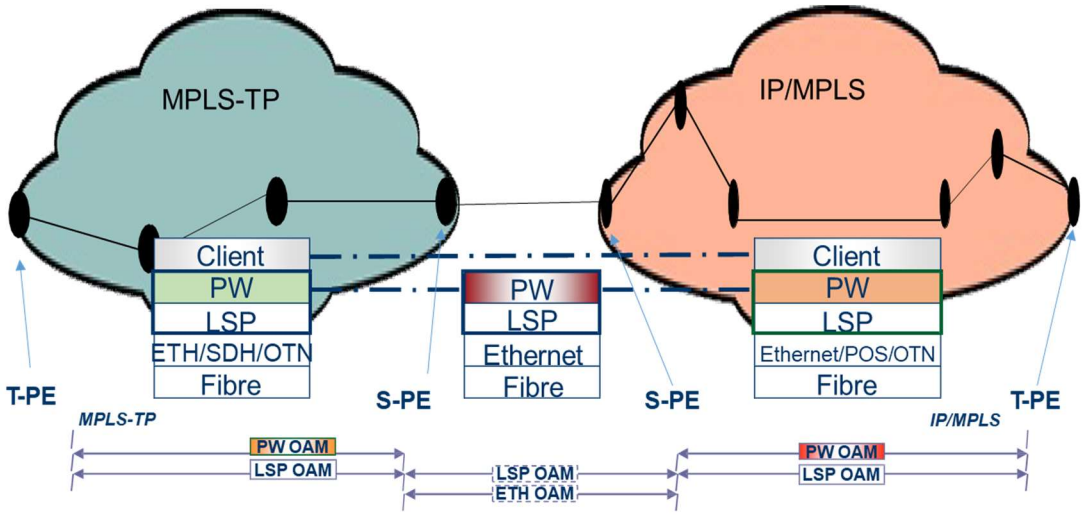


Figure 40 Modèle de cloisonnement proposé: « Segment de frontière »

Ce modèle utilise les mêmes mécanismes que le modèle précédent, à savoir les techniques MS –PW et « stitching LSP ». La seule différence est qu'il existe maintenant deux nœuds qui sont responsables de la segmentation du Pseudowire, se sont aussi deux nœuds frontières.

En ce qui concerne le modèle «segment de frontière », nous avons proposé une amélioration consistant à activer les fonctions d'auto-négociation d'OAM entre le S-PE de chaque segment. En effet, certains constructeurs d'équipement de transport proposent désormais de supporter les deux familles OAM. Notre proposition consiste à améliorer le modèle «segment de frontière » en permettant aux deux nœuds de frontière S-PE de négocier les options MIP OAM à utiliser. C'est l'objet de la section suivante.

4.3.4. Variante du modèle de cloisonnement « Segment de frontière » : ODM

Le modèle de référence OAM des réseaux MPLS-TP est basé sur certaines composantes fonctionnelles importantes : Entité de maintenance (ME), Groupe final de maintenance

(MEG), Surveillance de connexion en tandem (TCM), Points finaux MEG (MEP), Points intermédiaires MEG (MIP), et les MEP serveur, [45].

Le Domaine de Maintenance (MD) définit la partie du réseau qui est gérée et surveillée. Les MEP sont les limites du MD. Chaque MD possède un niveau ME de 0 à 7 afin de s'adapter à différents scénarios de déploiement. La MA permet de surveiller une instance de service au sein d'un MD, et chaque MA à l'intérieur du MD hérite de son niveau ME. Une MEP termine les trames OAM dans le MEG du niveau pour lequel elle est configurée.

La plupart des fonctions nécessaires à l'exécution de la couche de service OAM sont traitées entre les MEP. Quelques exigences pour configurer ces domaines de maintenance et leur association, les plus importantes sont : l'identifiant local et distant MEP Identifier MEP ID, l'identifiant MA (MAID) et le niveau MD. D'autres paramètres sont également nécessaires comme le taux d'intervalle CCM des messages de contrôle de continuité qui est utilisé pour les fonctions OAM proactives comme CC/CV, RDI et la mesure de perte de paquets.

Tout mécanisme de découverte d'OAM doit tenir compte de ces exigences. D'autres paramètres doivent être présents, représentés par des TLV (Type-Length-Value), pour pouvoir offrir des possibilités de négociation aux deux MEP. Grâce à ces TLV, les MEP peuvent choisir le type de norme OAM qui sera sélectionné. Nous suggérons également d'ajouter une nouvelle option TLV où les MEP peuvent utiliser l'authentification pour mettre en place une opération de maintenance. Cela peut s'avérer très utile lorsque plusieurs fournisseurs de services sont interconnectés pour offrir un service, en particulier pour les MEP qui se trouvent aux frontières du réseau.

Nous avons proposé un mécanisme de découverte d'OAM que nous avons baptisé ODM (OAM Discovery Mecanism)[65]. C'est est un mécanisme qui permet à deux MEP au sein d'un même MEG de négocier certaines options nécessaires à la mise en place des opérations de maintenance. Il évite l'approvisionnement manuel de l'ID du MEP distant et permet aux fournisseurs qui prennent en charge à la fois les normes MPLS-TP OAM (Y.1731 et BFD/LSP) de choisir la méthode OAM appropriée.

Tout comme le paquet MPLS-TP OAM, ODM peut être distingué des paquets utilisateur en utilisant les constructions G-ACh et GAL, précédemment développés dans le paragraphe « Opération, administration et maintenance » du 0. Il est également obligatoire que l'ODM soit supporté par chaque nœud membre d'une association d'opérations de maintenance.

La phase d'initialisation de la session ODM commence par l'envoi d'un message multicast de découverte à d'autres MEP distantes. Ce message de découverte contient des paramètres tels que l'ID de la MEP local, l'ID MEG, le niveau MD, le mot de passe si une authentification est activée, ainsi que le standard OAM pris en charge ou préféré. Les MEP distantes à l'intérieur d'un même MD et partageant le même niveau répondront au message de découverte par un message de réponse unicast avec chacun ses paramètres correspondants. La MEP ayant initié le processus de découverte ODM décide alors quelle norme OAM va être utilisée. Ainsi l'ensemble des MEP participantes posséderont à la fin du processus ODM les paramètres nécessaires pour établir une association d'opérations de maintenance, Figure 41.

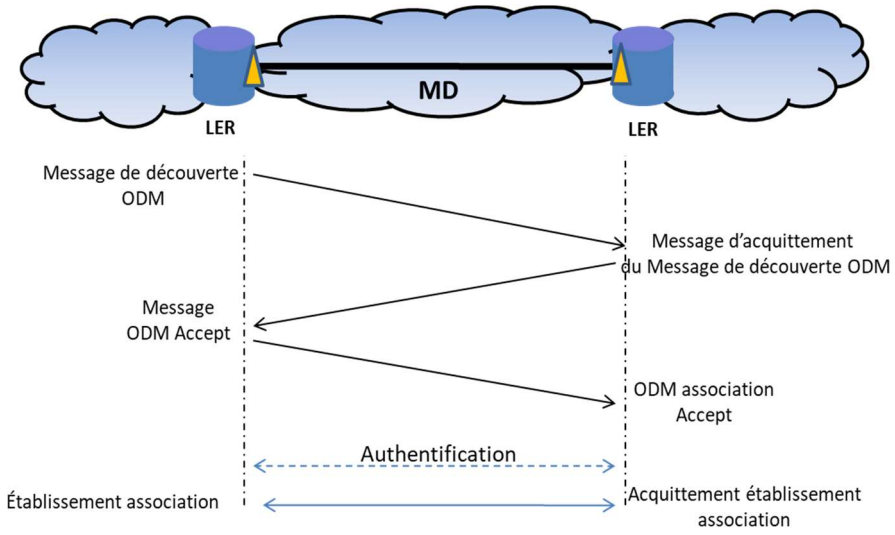


Figure 41 Processus d'établissement du mécanisme ODM proposé

Dans le cas où le processus de négociation échoue, une méthode OAM sera désignée par défaut. Cette méthode de repli doit être paramétrée au préalable. A l'issue de cette négociation, Figure 42, les éléments suivants sont échangés : Nom MD, Index MD, Index MA, Id du MEP et son adresse Mac.

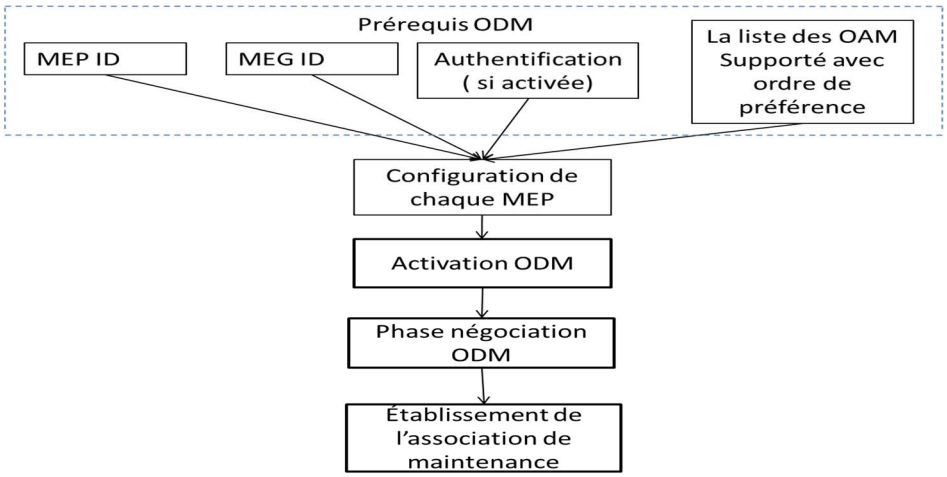


Figure 42 Déroulement du mécanisme ODM proposé

L'implémentation du module ODM nécessite que le software du LER, en l'occurrence le S-PE, soit ouvert et possède une API nord permettant à la fois de connaitre les options OAM, dont dispose le LER, et d'autre part de sélectionner le bon outil OAM suite à la convergence de la procédure de négociation ODM. Cet API n'est malheureusement pas disponible en ce moment dans la majorité des constructeurs LER. Par conséquent, on n'a pas eu la chance de tester notre proposition ODM. Bien entendu, le mécanisme ODM, combiné avec une approche de cloisonnement appropriée telle que le modèle « segment de frontière », permettra aux réseaux d'opérateurs de transport de fournir des services de qualité respectant des SLA très stricte moyennant une palette d'outils OAM très riche.

4.3.5. Evaluation des solutions proposées : simulations et résultats

Une maquette a été déployée pour valider le concept du modèle de cloisonnement statique. La maquette ainsi que les résultats sont présentés dans le paragraphe suivant.

a) Topologie de simulation :

Nous avons procédé à des tests en LAB pour pouvoir affirmer le bon fonctionnement du modèle proposé. Pour se faire, nous avons monté une maquette virtualisée permettant d'émuler un réseau de transport operateur composé de deux parties: la première en IP/MPLS et la deuxième en MPLS-TP. La plateforme utilisée pour la partie émulation est eve-ng, [63]. A noter que eve-ng est un émulateur de réseau qui prend en charge les images de routeurs commerciaux virtualisés (tels que Cisco et NOKIA) et les routeurs open-source. Il utilise Dynamips et IOS-on-Linux pour prendre en charge les images de routeurs et de commutateurs Cisco, et KVM/QEMU pour prendre en charge tous les autres dispositifs. Il est disponible sous forme d'image de machine virtuelle et peut également être installé sur un serveur dédié fonctionnant sous Ubuntu Linux.

Nous avons également opté pour un software Nokia SR7750 qui permettra d'émuler les nœuds IP/MPLS et MPLS-TP [64].

La topologie du réseau est composé d'un segment transporté sur du MPLS-TP et un autre sur du IP/MPLS, Figure 43, utilisant pour le premier la norme OAM ITU-T et pour le deuxième la norme OAM IETF. Un service E-line est fourni pour interconnecter le client CE1 au client CE2.

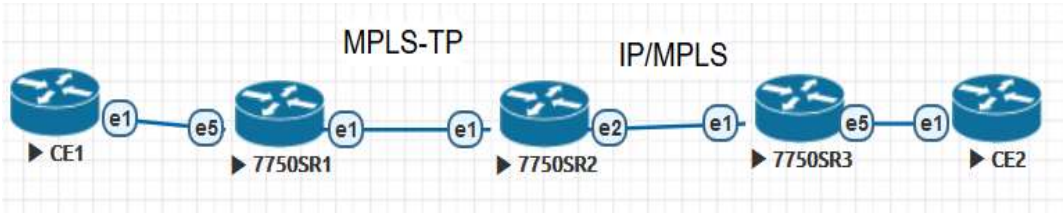


Figure 43 Schéma de la maquette du test de fonctionnement du modèle proposé

b) Déroulement des tests et analyse des résultats

Nous avons activé OSPF comme protocole de routage au niveau du réseau IP/MPLS ainsi que LDP comme protocole d'échanges de label MPLS. Concernant le réseau MPLS-TP, nous avons opté pour un « plan de contrôle » statique, donc aucun protocole de routage ni mécanisme d'échange de labels MPLS ne sera utilisé.

Ainsi, un service E-line démarre au niveau du PE 7750R1 avec une affectation statique de labels de service et pointant sur le PE 7750R2 qui joue un double rôle : il termine le segment MPLS-TP et démarre un autre segment en IP/MPLS. Le PE 7750R2 assure alors les fonctions d'Interworking des plans de contrôle et de gestion via la fonctionnalité MS-PW. Enfin, Le PE 7750R3 termine le segment IP/MPLS, Figure 44.

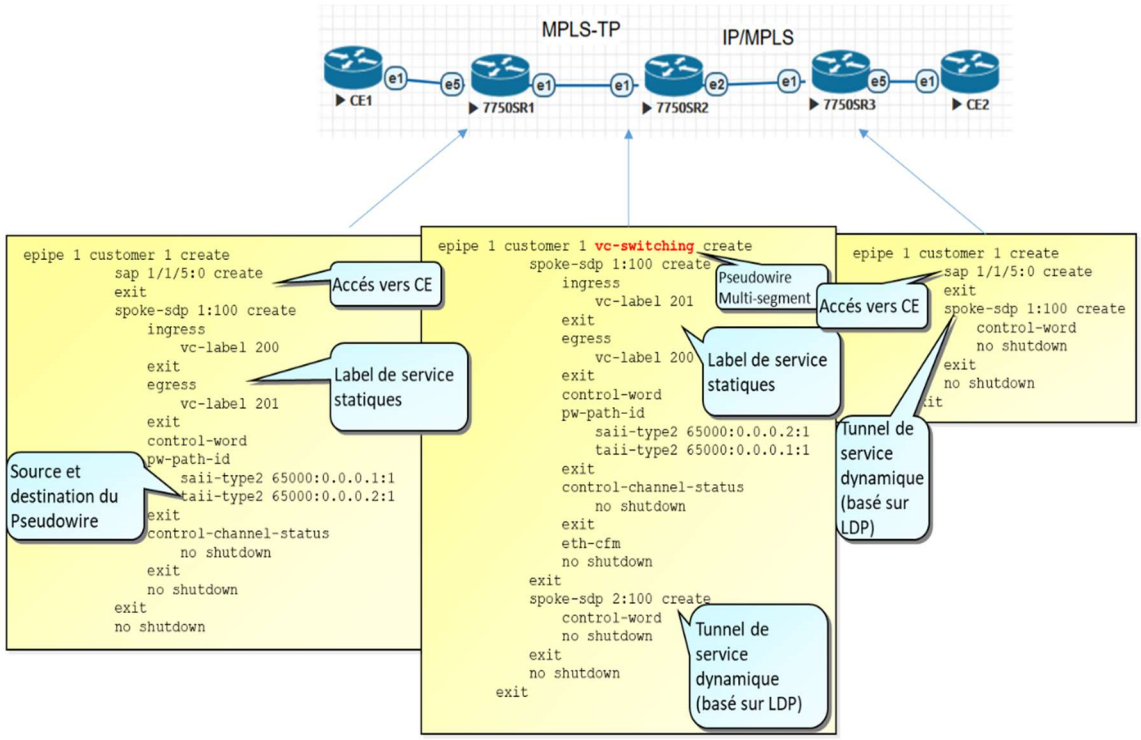


Figure 44 Configuration des éléments de la maquette

L'implémentation des outils OAM se fait à trois niveaux : LSP, PW et entre les CE.

La disposition des MEP au niveau du Pseudowire sont construites au niveau des PE d'extrémités du service E-line. Celle du LSP sont par contre construites à la fois sur le LSP MPLS-TP et aussi sur le LSP IP/MPLS.

On présente, dans la Figure 45, le positionnement des ME, MIP au niveau des segments de transport LSP et également au niveau du Pseudowire ainsi que leur configuration respectives.

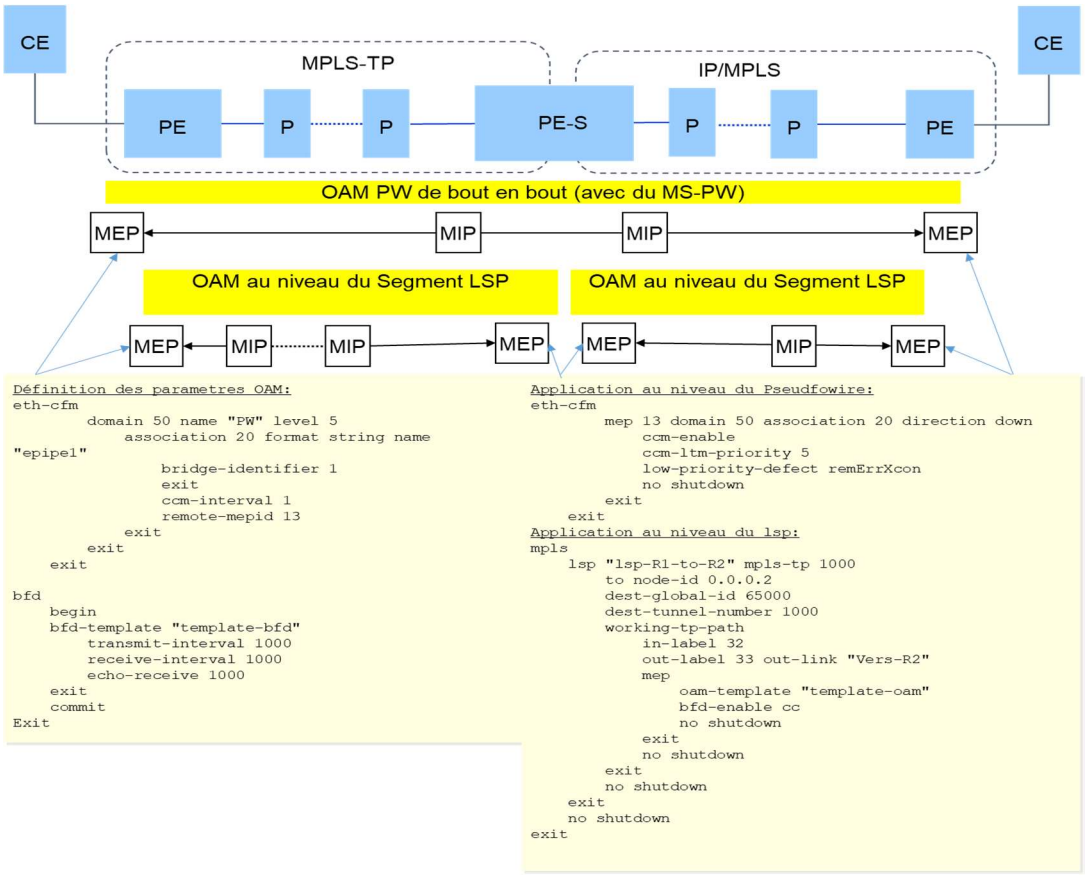


Figure 45 Construction des MEP et MIP dans le modèle de cloisonnement statique

Les résultats des traceroutes VCCV permettent de vérifier la continuité des messages OAM depuis le PE 7750R1 jusqu'au PE3 7750R3 en passant par PE2 7750R2 souligné en rouge sur cette Figure 46 :

```
*A:R1>config>eth-cfm# oam vccv-trace static 1:100 assoc-channel ipv4 detail
VCCV-TRACE 1:100  with 116 bytes of MPLS payload
1  10.0.0.2 GlobalId 65000 NodeId 0.0.0.2
   rtt=1.60ms rc=8(DSRtrMatchLabel)
   Next segment: VcId=100 VcType=Ether Source=10.0.0.2 Remote=10.0.0.3
2  10.0.0.3  rtt=3.76ms rc=3(EgressRtr)
```

Figure 46 Exemple de trace VCCV au niveau du pseudowire

Les tests de coupure d'un lien appartenant au chemin MPLS sur un segment sont propagés systématiquement sur l'autre segment du Pseudowire. Cette information est renvoyée comme message OAM via le Canal Générique Associé (G-ACh) présent au niveau du Pseudowire. Dans l'exemple suivant: nous avons coupé volontairement le lien entre les PEs R2 et R3, rendant ainsi en alarme le service E-line au niveau du PE R1.

```
*A:R1# show service id 1 all

===============================================================================
Service Detailed Information
===============================================================================
Service Id        : 1                 Vpn Id            : 0
Service Type      : Epipe
Name              : (Not Specified)
Description       : (Not Specified)
Customer Id       : 1                 Creation Origin   : manual
Last Status Change: 12/10/2018 07:11:33
Last Mgmt Change  : 12/09/2018 12:05:53
Test Service      : No
Admin State       : Up                Oper State        : Up
MTU               : 1514
Vc Switching      : False
SAP Count         : 1                 SDP Bind Count    : 1
Per Svc Hashing   : Disabled
Force QTag Fwd    : Disabled
-------------------------------------------------------------------------------
ETH-CFM service specifics
-------------------------------------------------------------------------------
Service Destination Points(SDPs)
-------------------------------------------------------------------------------
Sdp Id 1:100  -(0.0.0.2:65000)
-------------------------------------------------------------------------------
Description     : (Not Specified)
SDP Id            : 1:100               Type              : Spoke
Spoke Descr     : (Not Specified)
VC Type           : Ether               VC Tag            : n/a
Admin Path MTU    : 0                   Oper Path MTU     : 8914
Delivery          : MPLS
Far End           : 0.0.0.2:65000
Tunnel Far End    : n/a                 LSP Types         : MPLSTP
Hash Label        : Disabled            Hash Lbl Sig Cap  : Disabled
Oper Hash Label   : Disabled
Admin State       : Up                  Oper State        : Up
```

```
Acct. Pol          : None                  Collect Stats     : Disabled
Ingress Label      : 200                   Egress Label      : 201
Ingr Mac Fltr-Id   : n/a                    Egr Mac Fltr-Id   : n/a
Ingr IP Fltr-Id    : n/a                   Egr IP Fltr-Id    : n/a
Ingr IPv6 Fltr-Id  : n/a                   Egr IPv6 Fltr-Id  : n/a
Admin ControlWord  : Preferred                Oper ControlWord  : True
Admin BW(Kbps)     : 0                       Oper BW(Kbps)     : 0
BFD Template       : None
BFD-Enabled        : no                    BFD-Encap         : ipv4
Last Status Change : 12/09/2018 12:06:09     Signaling         : None
Last Mgmt Change   : 12/09/2018 12:05:53
Endpoint           : N/A                   Precedence        : 4
PW Status Sig      : Enabled
Force Vlan-Vc      : Disabled                Force Qinq-Vc     : Disabled
Class Fwding State : Down
Flags              : None
Local Pw Bits      : None
Peer Pw Bits       : lacIngressFault lacEgressFault
Peer Fault Ip      : None
Peer Vccv CV Bits  : None
```

Figure 47 Propagation d'OAM inter-segment d'un service E-line construit en MS-PW

Les flags **lacIngressFault et lacEgressFault** représentent un signalement d'alarme sur le segment IP/MPLS qui est en défaut.

Les tests réalisés sur maquette, Figure 40, ont été comparable aux tests réalisés avec la maquette du modèle de cloisonnement « nœud de frontière ». Sur un service E-line, l'implémentation des OAM de bout en bout s'est fait au niveau transport (LSP) et non au niveau service MS-PW. Les messages d'alarmes se propagent via les différents segments et les « lsp ping » ou « lsp tracroute » sont bien interprétés.

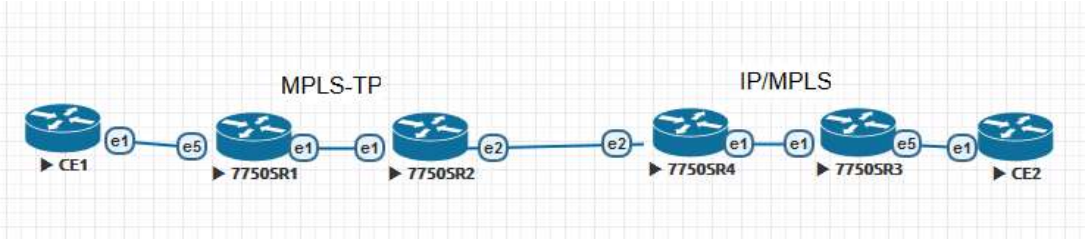


Figure 48 Schéma de la maquette: « Segment de frontière »

Les deux variantes du modèle de cloisonnement statiques, objet d'une publication [6], sont tout à fait appropriées pour résoudre les problèmes d'interopérabilité des outils OAM dans les réseaux de transport basés sur MPLS-TP.

Le modèle de cloisonnement « nœud de frontière » présente un avantage par rapport à celui du « segment de frontière » puisqu'on ne perd pas les fonctionnalités OAM PW «de « bout-en-bout » par contre il nécessite de supporter les fonctions d'IW au niveau du nœud d'interconnexion (entre les deux mondes OAM).

Les tests réalisés ont démontré que malgré le fait que le service de transport Ethernet E-line est segmenté en deux partie, une en IP/MPLS et une autre MPLS-TP, avec chacune une norme d'OAM différente, la solution de cloisonnement statique permet d'avoir une gestion de bout en bout. Cette solution est réalisable tant qu'on peut avoir un nœud supportant à la fois IP/MPLS et MPLS-TP.

4.4. Conclusion

Dans ce chapitre nous avons proposé des modèles de structuration des réseaux de transport qui permettent de répondre aux questions d'interopérabilité des outils OAM dans le milieu des réseaux de transport MPLS-TP. Le but recherche est de fournir une continuité OAM de bout en bout afin de pouvoir garantir la protection et la supervision des services transportés.

Le modèle « overlay » est le plus simple puisqu'il n y a aucune interaction « directe » entre les différentes couches. D'ailleurs, chaque couche « overlay », IP/MPLS ou MPLS-TP, est cliente de l'autre couche « Underlay » où l'unité de base de transport étant le Pseudowire PW. Toutefois, les entêtes supplémentaires ajoutés au niveau de la couche « Overlay », comme les labels de service et les labels de transport IP/MPLS, constituent un pourcentage important par rapport à la charge utile « Payload » du service transporté.

Le modèle de cloisonnement est la deuxième solution proposée. C'est une solution qui profite de technologies existantes élaborées pour le plan de contrôle, comme nous avons pu le de montrer, efficace pour le plan de gestion. En effet, d'après les résultats nos simulations, les techniques « MS-PW » et « LSP stitching » facilitent la définition de lignes de démarcation entre chaque domaine OAM. Les échanges au niveau des nœuds de frontières sont faites d'une manière naturelle permettant ainsi de répondre à l'un des critères les plus important dans les réseaux de transport MPLS-TP qui est celui de fournir des outils OAM de bout en bout. Ceci est dû au fait que les fonctions d'interfonctionnement (IW) sont primordiales dans ce genre de configuration.

Dans la deuxième variante du modèle de cloisonnement, « segment frontière», nous avons plutôt cherchée assure une continuité OAM au niveau transport. En effet, l'implémentation des OAM de bout en bout s'est fait au niveau LSP. Les messages OAM se propagent via les différents segments composant le service en question. L'option ODM est justement proposée pour palier à l'aspect manuel de l'implémentation de ce genre de modèle.

La solution de modèle « overlay » peut être comparée à celle de « couche fine » étudiée pendant l'état de l'art, elle a l'avantage de ne pas nécessiter de modification du software existant et tire profit uniquement des fonctionnalisées des réseaux MPLS-TP. Celle du modèle de « cloisonnement » peut être comparée à celle de « nœud d'interfonctionnement ». Il est vrai que les deux solutions proposent de s'appuyer sur l'IW

comme solution au problème d'interopérabilité, par contre notre solution présente l'avantage de n'utiliser l'option d'IW que sur un périmètre très restreint. En effet, grâce à la combinaison de la logique de cloisonnement et d'interfonctionnement, l'utilisation de l'option d'IW se fait uniquement au niveau du « nœud de frontière » contrairement à la solution « nœud d'interfonctionnement » qui peut faire appel à cette option sur plusieurs nœuds du réseau MPLS-TP.

Apres avoir traité les questions liées à l'interopérabilité des OAM, nous pensons que le plan de gestion doit prendre d'avantage d'importance dans les réseaux de transport. En effet, plusieurs opérateurs restent toujours septiques à l'idée d'avoir un plan de contrôle dynamique en activant des protocoles comme LDP, RSVP-TE ou autres protocoles de routage comme ISIS ou OSPF-TE. Rappelons que les réseaux de transport classiques ont toujours été gérés, d'un point de vue contrôle, à travers une station NMS, et que la configuration des circuits est faite d'une manière statique. C'est pour cette raison que nous proposons que les décisions prises au niveau du « plan de contrôle » ne soient pas uniquement le résultat de protocoles de routage mais aussi de pouvoir décider de changer un chemin LSP suite à une décision fournie par le plan de management : par exemple déplacement d'un seuil de délai de transit sur un LSP.

Dans les chapitres suivants, nous détaillons notre proposition qui utilise un système d'orchestration où les plans de contrôle et de gestion peuvent interagir sans pour autant altérer les normes et les standards déjà défini pour satisfaire les prérequis des réseaux de transport « nouvelle génération ».

Chapitre 5

Utilisation du concept SDN pour la gestion des OAM dans les réseaux MPLS-TP

On ne peut pas parler de réseaux « nouvelle génération » sans évoquer le sujet des « réseaux pilotés par logiciels » ou communément appelé SDN (Software Defined Networks). En effet, au moment même où on définissait les différentes briques de la technologie MPLS-TP dans les coulisses de l'IETF et de l'ITU-T, l'Open Networking Foundation ONF a mis en point les principaux traits du standard SDN [66].

Il existe, toutefois, beaucoup de questions sur la manière avec laquelle les outils OAM seront utilisés et surtout comment tirer profit de ce nouveau paradigme SDN d'où l'idée de développer un concept d' « OAM-as-a-Service ».

Dans ce chapitre, on commence par une brève présentation des composantes SDN. Par la suite, on détaille notre solution basé sur le concept SDN pour la gestion des OAM dans les réseaux MPLS-TP.

5.1. Présentation du paradigme SDN

5.1.1. Introduction

L'avènement d'Internet a conduit le monde qu'on connaissait à devenir une société numérique où presque tout est connecté et accessible depuis partout. Victime de leurs succès, les réseaux informatiques traditionnels sont devenus très complexes et difficiles à gérer [67]. A l'heure actuelle, les réseaux traditionnels ne disposent pas de mécanismes leurs permettant une auto-reconfiguration. Dans la Figure 49 on voit que l'architecture des réseaux actuels est divisé en trois parties: un plan de contrôle concerné par déterminer les règles à suivre pour prendre en charge un trafic réseau, un plan de données en charge de transmettre le trafic suivant les règles définis par le plan de contrôle, et un plan de gestion incluant des services logiciels. Cette architecture est implémentée sur chaque équipement réseau, réduisant ainsi la flexibilité à innover et à faire évoluer les infrastructures réseaux.

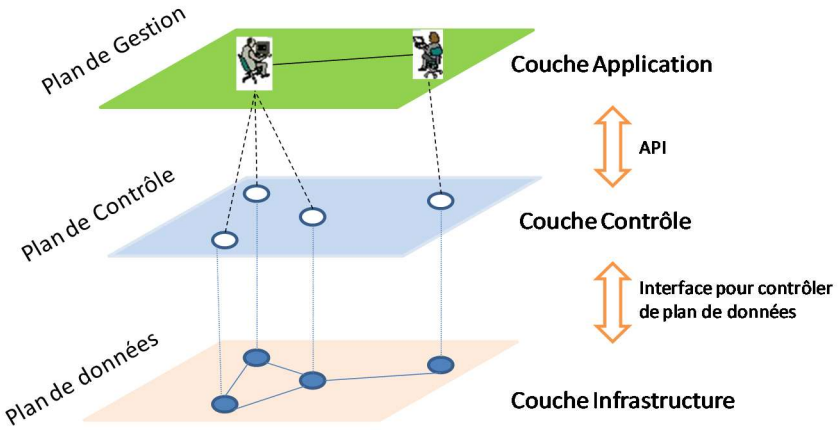


Figure 49 Vue sous forme de couches du paradigme SDN

Le SDN est un nouveau paradigme d'architecture réseau qui consiste à découpler les plans de contrôle et de données des équipements réseaux qui, jusque-là, cohabitaient ensemble dans le même équipement. Le résultat de ce découplage est d'avoir deux types de composantes au niveau réseau : Un équipement avec uniquement un plan de donnée appelé commutateur SDN et un autre élément en charge du plan de contrôle, appelé contrôleur SDN [68]. Toutefois, Il est important de souligner que ce modèle logique de programmation centralisée n'implique pas un système physiquement centralisé [69]. En effet, la nécessité de garantir des niveaux élevés de performances, d'évolutivité, et de fiabilité empêcherait une telle supposition et nécessite souvent le recours à une architecture physiquement distribuée au niveau du plan de contrôle [69], [70]. Cette séparation du plan de contrôle et du plan de données peut être réalisée au moyen d'une interface de programmation bien définie entre les commutateurs et leur contrôleur SDN puisque ce dernier exerce un contrôle direct sur l'état du plan de données via une interface de programmation d'application API. Openflow est l'exemple d'API les plus remarquables [71], [66].

5.1.2. Architecture SDN

L'architecture SDN est composée de trois différents plans, comme décrit dans la Figure 50, qui sont : plan applicatif, plan de contrôle et plan de donnée.

Plusieurs fonctions spécifiques permettent l'interaction entre ces différents plans:

- l'interface Sud (Southbound API) : qui sert de connexion entre les plans de contrôle et celui de données, permettant ainsi la séparation des fonctionnalités de ces deux plans.
- le Système d'exploitation réseaux : qui est responsable de la fonction de contrôle permettant de générer la configuration réseau basé sur les règles définies par l'opérateur de réseau. Il permet également de fournir les abstractions, les services réseau essentiels, et les interfaces de programmation d'application commune (API) aux développeurs.

- l'interface Nord (Northbound API) : qui permet d'offrir une abstraction entre les applications des développeurs d'une part et les détails internes des fonctions du contrôleur et du comportement du plan de données d'autre part.
- les applications réseau : qui mettent en œuvre une logique de contrôle capable de traduire les commandes installées au niveau du plan de données dictant ainsi le comportement des commutateurs SDN.

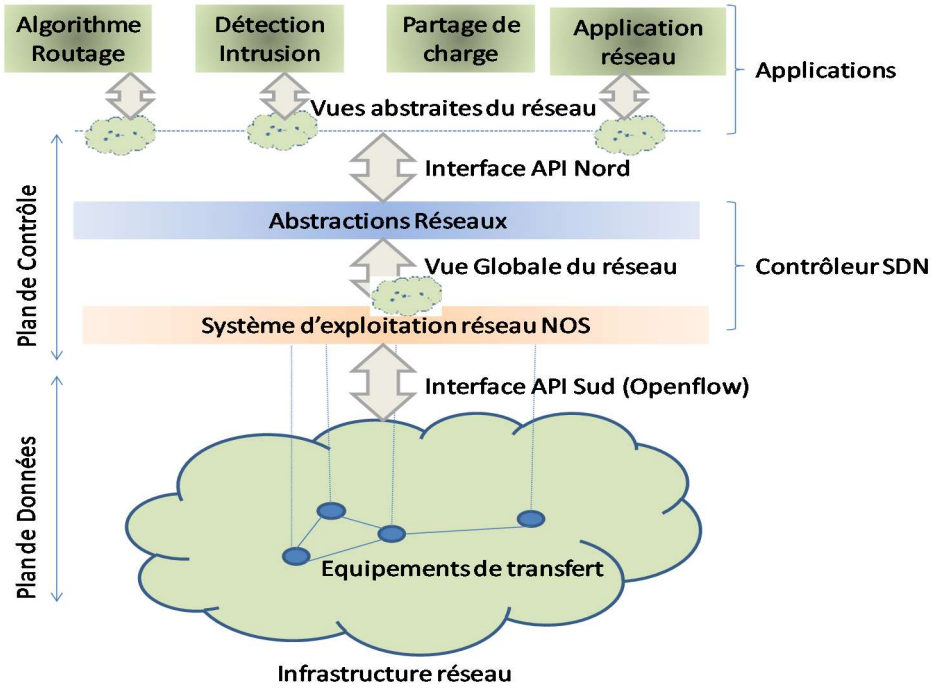


Figure 50 Vue globale sur l'architecture SDN

Dans les sous sections suivantes on présentera deux principales composantes qui sont le protocole Openflow et les commutateurs Openflow.

a) OpenFlow

Le protocole OpenFlow constitue un élément fondamental et indispensable dans la conception des solutions SDN. Il est, en tant que API Sud, le protocole le plus utilisé dans les implémentations SDN. La « Fondation des réseaux Ouvert » (ONF) s'est chargée de définir le mode de fonctionnement et les spécifications du protocole Openflow [72].

Dans un routeur ou un commutateur classique, la transmission ou transfert de paquets (lié au plan de données) et les décisions de routage (lié au plan de contrôle) sont exécutées sur le même équipement. Par contre, ces deux taches sont séparées dans un commutateur Openflow, ainsi, le plan de données réside toujours sur le commutateur, tandis que les décisions de routage de haut niveau sont déplacées vers un élément de contrôle séparé. Le commutateur et son contrôleur communiquent via le protocole OpenFlow. Ce dernier définit les messages de contrôle, tels que des paquets reçus, les paquets à commuter, la modification de la table de commutation, et l'obtention des statistiques du commutateur.

Le plan de données d'un commutateur OpenFlow présente une table d'abstraction de flux où chaque entrée du tableau de flux contient un ensemble de champs des paquets associée

à une action ou à plusieurs actions. Quand un commutateur OpenFlow reçoit un paquet pour lequel aucune entrée équivalente n'existe dans la table des flux, il envoie ce paquet au contrôleur. Ce dernier décide alors du traitement à donner ce paquet, et en même temps d'ajouter, de supprimer ou de modifier une entrée dans la table de flux. Ce genre de décisions permettra au commutateur d'appliquer, à l'avenir, le même traitement des paquets similaires et d'éviter ainsi de revenir au contrôleur à chaque réception de paquet sur le commutateur.

Ainsi selon les règles prescrites par le contrôleur SDN, un commutateur Openflow peut se comporter comme routeur, commutateur Ethernet ou MPLS, Pare-feu ou avoir un autre rôle comme élément de partage de charge « Loadbalancer » ou translateur d'adresse IP, etc…

b) Commutateur Openflow

Un commutateur OpenFlow se compose d'une ou plusieurs tables de flux (Flow Table) en pipeline et d'un groupe de table, Figure 51. L'ensemble de ces tables forment le support pour l'identification et le transfert des paquets. Le commutateur OpenFlow dispose également d'un canal sécurisé dédié aux échanges avec le contrôleur externe.

Chaque table de flux contient un ensemble d'entrés de flux qui décrivent la manière avec laquelle un paquet, reçu au niveau du commutateur SDN, va être traité. Le Tableau 5 présente la composition d'une table de flux :

Champ de correspondance	Priorité	Compteurs	Instructions	Timeouts	Cookie	Flag

Tableau 5 Composantes d'une table de flux

Une entrée de flux est composée des champs suivants:

- un ensemble de champs de correspondance (matching field) qui définissent le modèle du flux de paquets à travers l'instanciation des champs d'entête allant de la couche Ethernet à la couche Transport,
- des compteurs et des statistiques sur les paquets
- des instructions qui indiquent les décisions à prendre à travers le pipeline
- et des actions à appliquer sur le paquet qui correspondent à l'entrée en question.

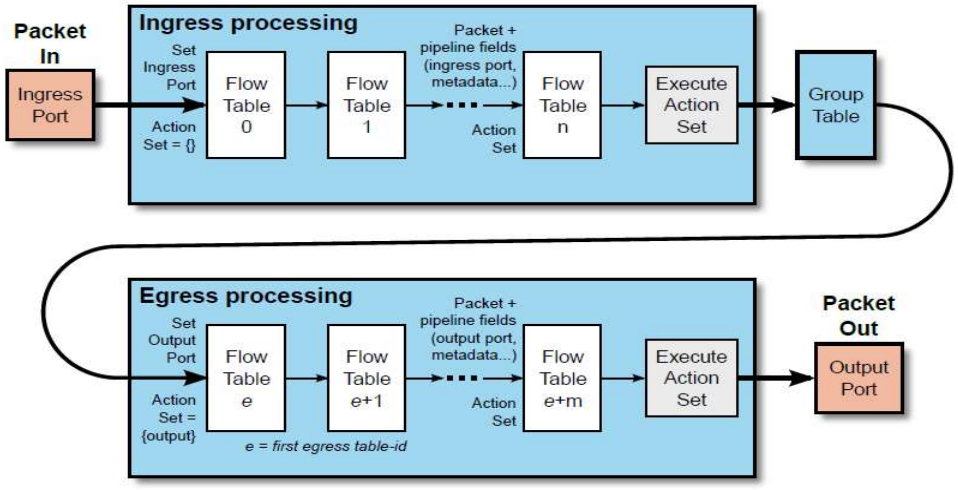


Figure 51 Parcours d'un paquet au niveau du pipeline

Le processus d'identification commence par vérifier la Table 0 qui est la première table du pipeline et peut continuer son traitement au niveau d'autres tables additionnelles. Les entrées de flux sont appliquées dans un ordre de priorité, ainsi la première entrée sera utilisée pour traiter le paquet en question. Si une correspondance est trouvée, alors les instructions, associées à l'entrée en question, seront exécutées. Si aucune correspondance n'est trouvée, la suite du processus dépendra de la configuration par défaut du commutateur OpenFlow. Il existe trois cas de figure: le paquet peut être envoyé au contrôleur, supprimé, ou passé à la table de flux suivante.

Il existe plusieurs types de champs de correspondances, Figure 52, supportés dans OpenFlow: port d'entrée, métadonnées, adresse MAC source, adresse MAC destination, type Ethernet, identifiant VLAN, priorité VLAN, label MPLS, classe de trafic MPLS, adresse IPv4 source, adresse IPv4 destination, protocole IPv4, type de service IPv4, port source TCP/UDP/SCTP ou type ICMP, port destination TCP/UDP/SCTP ou code ICMP. Chaque entrée de flux contient des valeurs spécifiques de champs de correspondance qui permettent de les comparer avec l'entête du paquet ainsi que son numéro de port d'entrée. Les instructions associées à chaque entrée flux décrivent les modifications et le transfert du paquet, le traitement sur la table de groupe et le traitement sur le pipeline des tables. Le traitement au niveau du pipeline permet aux paquets d'être envoyés aux tables suivantes pour un traitement supplémentaire et permet à des informations sous forme de métadonnées d'être échangées entre les tables. Le traitement du pipeline s'arrête quand l'ensemble des instructions d'une entrée flux ne spécifie pas de table suivante. A ce stade, le paquet est généralement modifié et transmis.

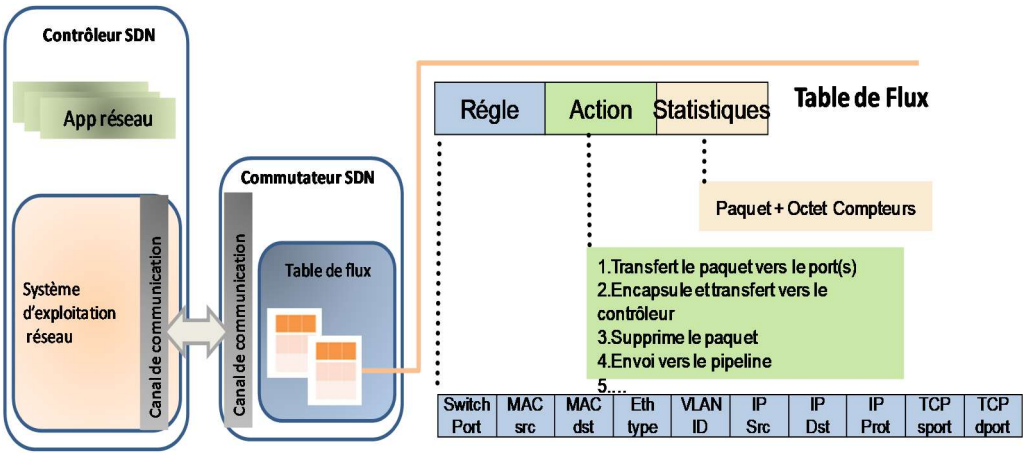


Figure 52 Les équipements de transfert SDN fonctionnant avec OpenFlow

Le protocole OpenFlow dispose de cinq types d'instructions: appliquer des actions, effacer les actions, écrire des actions, écrire des métadonnées et aller à la table suivante. Il s'agit d'actions qui seront appliquées sur le paquet et qui sont classées selon deux catégories: des actions de type push/ pop permettant de faire l'encapsulation et la décapsulation des paquets MPLS et des trames de VLAN, et des actions de type « positionner-un-champ » (set-field) permettant de modifier la valeur d'un champ d'un paquet en allant de la couche 2 jusqu'à la couche 4 du modèle OSI.

5.2. Adoption du Paradigme SDN dans les réseaux MPLS-TP

Les réseaux traditionnels sont complexes et difficiles à gérer, et pour cause la manière verticale et dépendante de chaque constructeur (ou version) avec laquelle les plans de contrôles et de données sont intégrés. En d'autres termes, chaque ligne de produit peut avoir ses propres interfaces de configuration et de gestion particulières, ce qui implique de longs cycles de production ou de mises à jour rendant ainsi les changements et l'innovation très compliqués.

Plusieurs travaux ont été menés dans le but de faciliter l'innovation dans les réseaux moyennant un « plan de contrôle » centralisé permettant ainsi de simplifier les changements. Cette vision est portée notamment par une technologie basée sur le PCE de « Path Compute Element » qui est un élément de contrôle permettant le calcul des chemins sur un réseau dont les éléments PCC (Client du PCE). Le PCE communique avec les PCC en utilisant le protocole PCEP (PCE Communication Protocol)[57].

En particulier, l'article [73] propose l'utilisation d'un PCE pour contrôler et trouver les chemins LSP sur un réseau MPLS-TP. Cette architecture présente l'avantage de pouvoir développer des applications ou services réseaux au niveau du serveur PCE. Toutefois, le PCE ne peut agir que sur le plan de contrôle et donc est limité en termes d'interaction avec le plan de gestion. Nous avons donc adopté ce même concept et avons fait le parallélisme avec le paradigme SDN pour proposer une solution permettant de contrôler le réseau MPLS-

TP utilisant ainsi un contrôleur SDN et le protocole Openflow pour communiquer avec les commutateurs MPLS-TP.

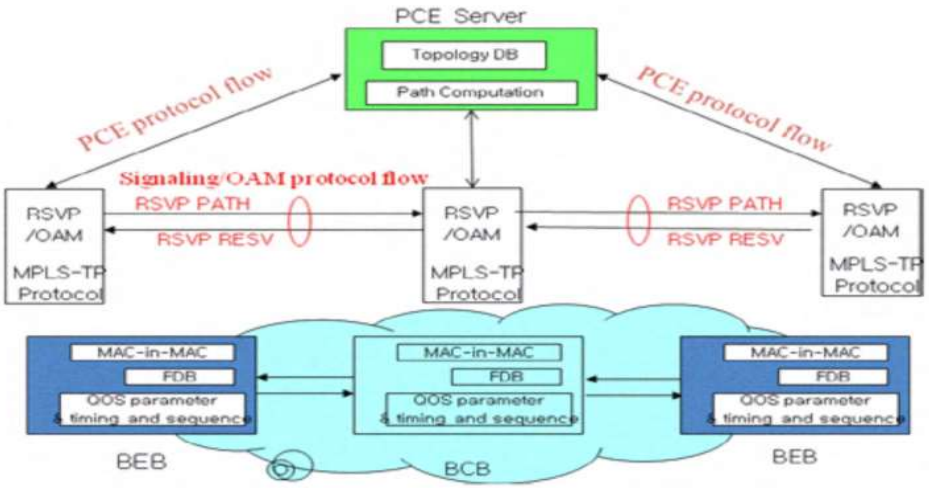


Figure 53 Utilisation du PCE pour contrôler un réseau MPLS-TP

En effet, le paradigme SDN crée une opportunité pour résoudre ces problèmes de longue date. En effet, le fait que SDN reste transparent par rapport au plan de données des réseaux de transport constitue un élément clef pour l'appliquer sur MPLS-TP. Dans le **Tableau 6** nous listons d'autres caractéristiques permettant d'appliquer le modèle SDN aux réseaux de transport MPLS-TP.

	MPLS-TP	SDN
Modèle de Service	Point à Point, L2VPN	Point à Point, L3VPN, L2VPN
Orienté Transport	Oui	Oui
Mécanisme de Transport	PseudoWire sur LSP	Openflow
Plan de données	MPLS	Variété de protocoles (y compris MPLS)
Plan de Contrôle	Statique(NMS), GMPLS	OpenFlow
Support de la QoS	E-LSP/L-LSP	Finement gradué

Tableau 6 caractéristiques SDN et MPLS-TP

Compte tenu de cette vision plus globale du réseau, le contrôle centralisé permet de facto de résoudre les problèmes d'interopérabilités puisque c'est le rôle du contrôleur de dicter quel standard OAM est à utiliser pour tous les nœuds du réseau MPLS-TP.

Notre idée est alors d'appliquer la paradigme SDN dans l'objectif, d'avoir d'abords un plan de contrôle centralisé possible de programmer, et bien entendu de démontrer qu'il est tout à fait possible de lancer des outils OAM en tant que service réseau. Ces services réseau OAM doivent être lancés depuis l'API nord de notre contrôleur SDN.

Ainsi pour illustrer ce concept, nous avons développé une méthode OAM comme « preuve de faisabilité » (Proof Of Concept). Cette méthode permet de mesurer le délai aller-retour de bout en bout sur un chemin LSP au niveau d'un réseau MPLS-TP.

Dans la suite de ce chapitre, on présentera cette solution OAM mesurant le délai de transit aller/retour RTT (round Trip Time) entre deux point d'un PW sur un réseau MPLS-TP en se basant sur le concept SDN, [74].

5.3. Implémentation d'une solution de mesure de délai basée sur SDN

Dans notre approche, les LSR et LER représentant les nœuds du réseau MPLS-TP feront aussi fonction de commutateur Openflow. Ces commutateurs implémentent les règles de forwarding, constituant le « plan de donnée », via les messages Openflow envoyé depuis le Contrôleur Openflow. Ce dernier communique avec d'autres modules, via son interface API nord, qui chacun peut avoir une fonction comme le routage, le système de provisioning des services, ou comme notre cas un module OAM de mesure de délai RTT.

5.3.1. Fonctionnement du module « Mesure de délai » RTT sur un modèle SDN

La méthode proposée pour mesurer le délai est basée sur l'injection d'un paquet de sonde provenant du contrôleur, puis envoyé au commutateur source qui l'enverra au commutateur de fin de course par le chemin de commutation de l'étiquette utilisateur, et enfin retournera au contrôleur.

Sur la base des normes relatives et des recommandations des exigences MPLS-TP, les paquets OAM devraient fonctionner in-band au même titre que les paquets de données. Ainsi, le paquet écho OAM généré par le contrôleur doit prendre le même chemin que l'utilisateur, Figure 54.

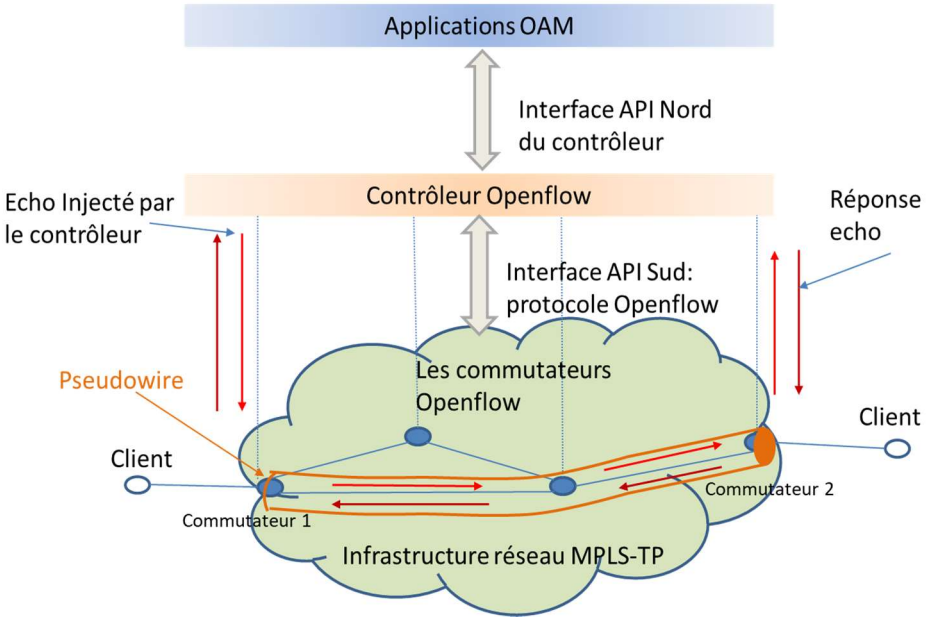


Figure 54 Module OAM pour le calcul de délai de transit sur un chemin

5.3.2. Concept du module OAM proposé

Afin d'élaborer un module capable de mesurer le délai, nous avons besoin de savoir quels messages sont échangés entre les commutateurs et le contrôleur.

Après avoir démarré notre topologie comprenant le contrôleur, les hôtes et les commutateurs, nous avons défini des valeurs par défaut relatives à chaque lien, en particulier la bande passante, la perte et le délai. Une fois que les commutateurs Openflow établiront la connexion tcp avec le contrôleur, celui-ci répond par un message Openflow de type Features_Request ce qui permet à la fonction EventOFPSwitchFeatures de remplir les tables de flux dans les commutateurs 1, 2 et dans les commutateurs intermédiaires. Les commutateurs intermédiaires exécuteront principalement des opérations de swap de labels MPLS, tandis que les commutateurs 1 et 2 devront effectuer des opérations de push et de pop de labels MPLS. Les tables de flux commutateurs 1 et 2 devraient être capables de traiter trois types de paquets :

- Paquets provenant d'hôtes Clients (p. ex. protocole de résolution d'adresses ARP, trafic IP normal avec ou sans étiquette VLAN)

- Paquets provenant de "Network" : à partir d'autres commutateurs (par ex. étiquetés avec MPLS Label dans notre cas)

- Les paquets provenant du contrôleur, y compris les paquets OAM d'écho et de réponse écho utilisé pour calculer le délai de transit RTT.

C'est la raison pour laquelle les tables de flux doivent être initialement installées sur les commutateurs afin de prendre en compte les différents types de flux.

Une fois les tables de flux installées, le contrôleur commence à envoyer des requêtes de statistiques à partir des commutateurs afin de calculer leurs RTTs associés en cochant l'événement Statistiques EventOFPPortStatsReply.

En parallèle, un autre processus de surveillance envoie en continu un message d'écho OAM du contrôleur à l'adresse mac source et destination en les substituant, utilisant le mécanisme de spoofing arp [75], respectivement sur l'adresse source mac du port source du commutateur1 connecté au client1 et sur l'adresse destination mac du port du commutateur2 connecté au client. Le gestionnaire d'événements Packet_In permet alors de connaître l'heure d'arrivée du paquet au contrôleur.

Hormis le délai de transit entre deux commutateurs représentant les points d'entrée et de sortie du service Pseudowire, d'autres valeurs de délai sont à prendre en considération notamment les délais de communication entre contrôleur et commutateur, mais aussi les différents délais de processing sur le contrôleur lui-même et également sur les commutateurs.

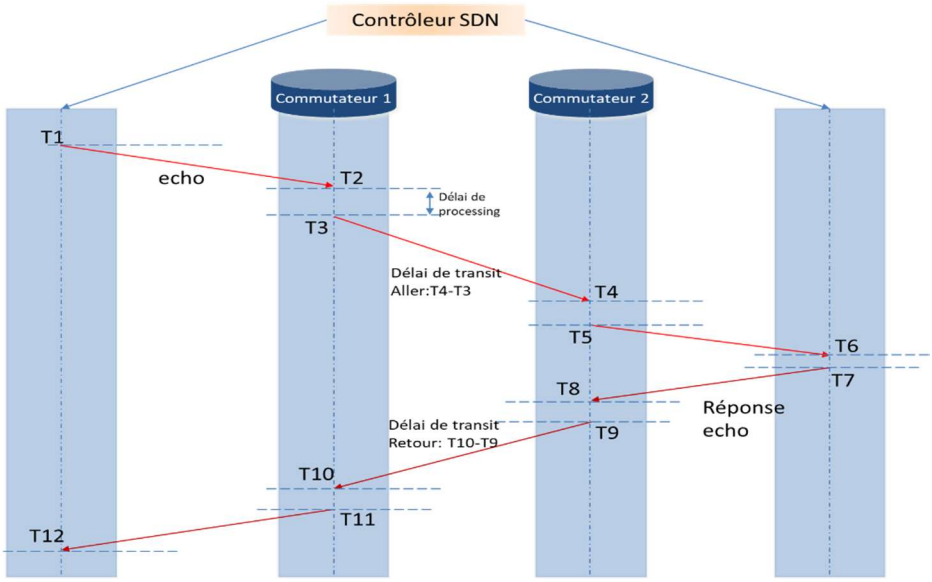


Figure 55 L'ensemble des délais impliqués dans le calcul du délai de transit aller-retour

Dans la Figure 55 on retrouve l'ensemble des délais impliqués dans le calcul du délai de transit aller-retour sur un pseudowire. La formule du délai de transit RTT est :

Trtt = (T10-T9) + (T4-T3)

Ou le délai total d'un paquet echo est égal à : **Ttotal =T12-T1**

Or nous disposons au niveau contrôleur que des valeurs : T1, T6, T7 et T12

Le délai de processing étant de l'ordre de microsonde, et est infiniment petit par rapport au délai de transit qui est de l'ordre de millisecondes. On va donc supposé qu'il est égal à zéro.

Et donc en simplifiant les valeurs : T2=T3, T4=T5, T6=T7, T8=T9, et T10=T11

Trtt = Ttotal – [(T2-T1) + (T12 –T11)] – [(T6-T5) + (T8-T7)]

La valeur [(T2-T1) + (T12 –T11)] est le temps de transit aller/retour entre le contrôleur et le commutateur sur un canal openflow. On supposera que les valeurs de délai dans le sens commutateur vers contrôleur et dans le sens contrôleur vers commutateurs sont égales.

La formule de calcul de délai de transit sur un pseudowire RTT aller/retour devient plus simplifiée :

Trtt = Ttotal – Trtt-commutateur1 – Trtt-commutateur2

Cette formule ne dépend pas du temps de synchronisation entre les nœuds (entre les commutateurs et le contrôleur, ou entre les commutateurs entre eux) puisque toutes ces variables dépendent uniquement de l'horloge du contrôleur. Il n'est donc pas nécessaire d'avoir un serveur NTP dans notre banc d'essai.

Cette formule ne tient pas compte du retard causé par le traitement des nœuds. Ceci est dû au fait que nous utilisons des machines virtuelles pour simuler des hôtes, des commutateurs et des contrôleurs.

Ainsi, nous sommes en mesure de connaître le délai total qu'il a fallu au message de la sonde pour passer par tous les commutateurs de ce chemin. En faisant correspondre le couple d'adresse mac source de commutateur1 et d'adresses mac de destination de commutateur2, nous pouvons calculer le délai total pour un chemin de bout en bout.

Cette logique est transcrite ci-dessous, bien entendu nous avons exécuté au préalable les fonctions permettant la construction de la topologie du réseau, les services pseudowires et l'injection des paquets OAM écho et écho-reply.

```
#Mesure de RTT entre commutateur 1 & 2 et Controlleur
if evenement EventOFPPortStatsReply
if datapath-id = commutateur1
        #mesure RTT1
                RTT1=received_time - sent_time1
              #measure RTT2
        elif datapath-id = commutateur2
              RTT2= received_time - sent_time1
#Mesure de des timers T6 et T12
elif evenement EventOFPPacketIn
                        if source-mac= Controlleur && packet-type = echo
                        T6 = time.time() * 1000 - T1
                        elsif source-mac= Controlleur && packet-type = echo-replay
                        T12 = time.time() * 1000 -T1
```

L'organigramme ci-dessous, Figure 56, explique comment les paquets sont traités au niveau du contrôleur et du commutateur. En effet, un module « Packet_in_handler » prend en charge tous les messages envoyé depuis les commutateurs Openflow vers le contrôleur. On isole le trafic OAM et en fonction la nature des paquets, on prend la décision d'effectuer les opérations de push ou de pop du label MPLS pour que nos flux OAM générés soient « in-band » et envoyés ainsi dans le plan de donnée. Au niveau du PE de livraison, on isole le flux OAM pour être re-envoyer au contrôleur. Le calcul du délai de transit peut ainsi être calculé.

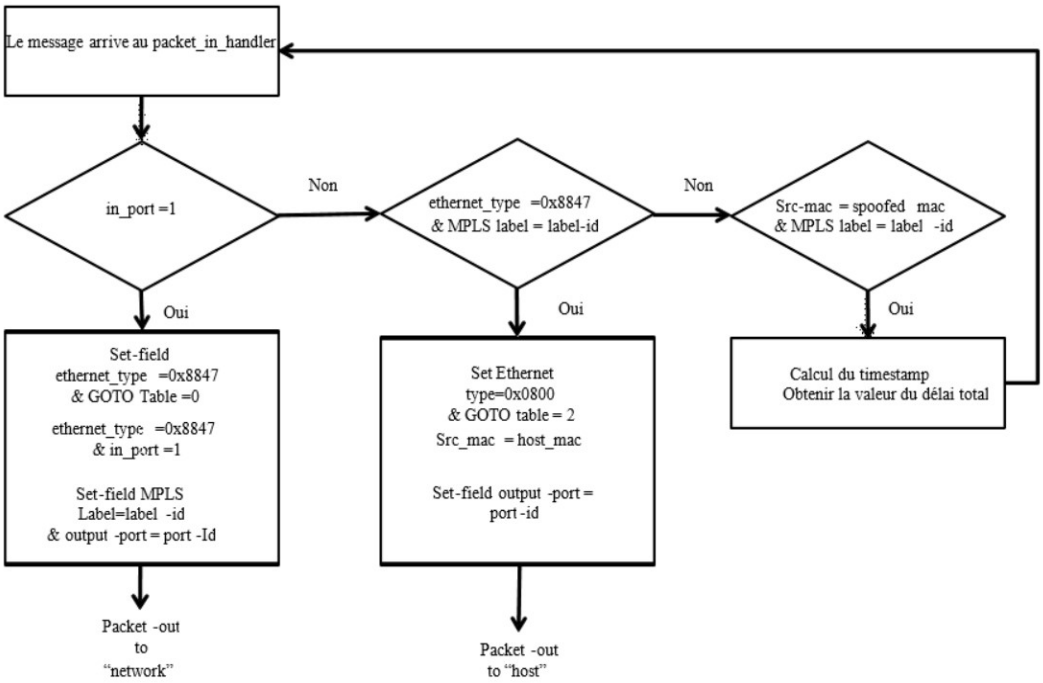


Figure 56 règles et actions de chaque table de flux du module OAM

5.3.3. Evaluation de la solution proposée

Dans cette section, nous présentons l'environnement proposé pour notre prototype avec les différents scénarios d'essais.

a) Le banc d'essai

L'émulation utilise deux ordinateurs portables Intel Core i5 avec quatre cœurs 2,3 Ghz et 4 Go de RAM. La première machine virtuelle émule le contrôleur tandis que la seconde émule les commutateurs et les hôtes, Figure 57.

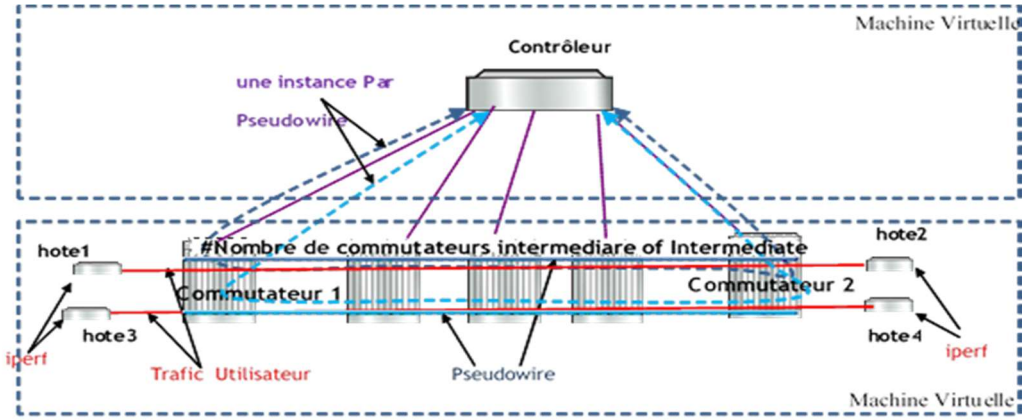


Figure 57 Maquette de simulations

Nous utilisons deux logiciels libres pour construire notre prototype :

- Ryu : qui joue le rôle du contrôleur Openflow, [76].

- Ofsoftswitch : qui fonctionne en tant que commutateur Openflow, [77].

Nous utilisons également Mininet qui est un émulateur réseau capable d'émuler les hôtes (clients) et les différents liens. Mininet prend en charge différents types de commutateurs Openflow. Nous avons choisi Ofsoftswitch13, [78], car il s'agit d'une implémentation logicielle d'espace utilisateur compatible Openflow 1.3 supportant MPLS et les fonctionnalités statistiques.

Les hôtes génèrent du trafic UDP en utilisant la commande iperf Linux, qui est un outil de mesure active de la bande passante maximale possible sur les réseaux IP, (ESnet). Il prend en charge le réglage de divers paramètres liés à la synchronisation, aux protocoles et aux tampons. Pour chaque test, on notera la bande passante, la perte de paquets et d'autres paramètres.

Le module OAM est développé à l'aide de l'API Ryu, où le contrôleur interroge périodiquement les commutateurs à un intervalle constant pour recueillir des informations sur les délais. La topologie du banc d'essai est également personnalisée à l'aide de l'API Mininet. Nous utilisons la version Ryu 3.13 et la version mininet 2.1.0+.

Nous avons exécuté deux scénarios avec 8 différents débits à chaque fois:

- Contrôleur exécutant une instance du module OAM pour un pseudowire. Le nombre de commutateurs intermédiaires est augmenté au fur et à mesure en même temps iperf est lancé sur les hôtes pour générer du trafic à différents débits.

- Contrôleur exécutant plusieurs instances du module OAM pour plusieurs pseudowires. Le nombre de commutateurs intermédiaires est augmenté au fur et à mesure en même temps iperf est lancer sur les hôtes pour générer du trafic à différents débits.

La mesure retenue est la moyenne des mesures prises pendant un intervalle de 100 secondes.

L'intervalle d'envoi de message OAM écho est réglé à 500 ms, on l'appellera l'intervalle de probing. Les valeurs de probing inférieures à 500 ms ont donné lieu à un comportement imprévisible avec un retard très excessif. Ceci est dû au fait que les processus de Mininet ne fonctionnent pas en parallèle, mais qu'ils utilisent le multiplexage temporel.

Les hote2 et hote4 exécutent iperf comme serveur, Figure 58:

```
h2.cmd( 'iperf -s –i 1 -u &' )
h4.cmd( 'iperf -s –i 1 -u &' )
```

Où "-s" signifie serveur et "-u" signifie UDP.

Les hote1 et Hote3 génèrent du trafic UDP en utilisant iperf :

```
h1.cmd( 'iperf -c ' + h2.IP() + ' -u -t 100 -b 1M >> h1-1M.log &')
h3.cmd( 'iperf -c ' + h4.IP() + ' -u -t 100 -b 1M >> h4-1M.log &')
```

Où "-t" signifie temps réglé à 100 secondes et "-b" signifie de bande passante réglée à 1Mbps.

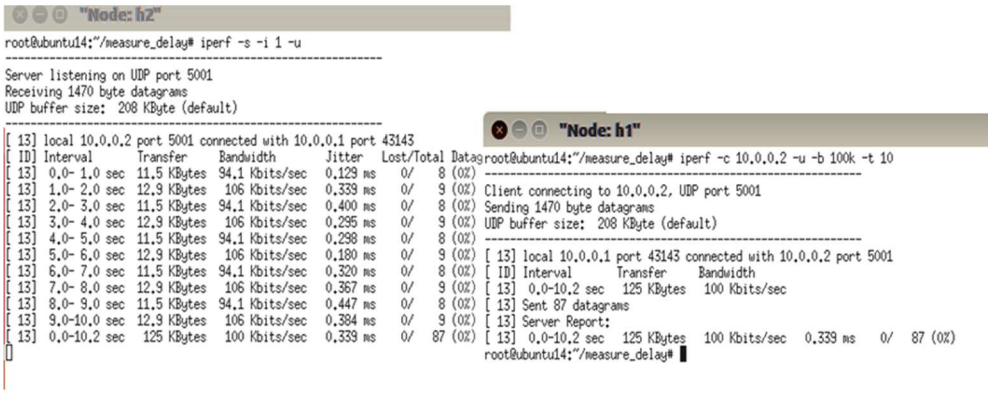


Figure 58 iperf serveur et client

Nous avons commencé notre premier test avec une topologie simple afin d'effectuer l'étalonnage et de connaître la valeur du délai lorsqu'il n'y a pas de trafic sur le réseau, Figure 59.

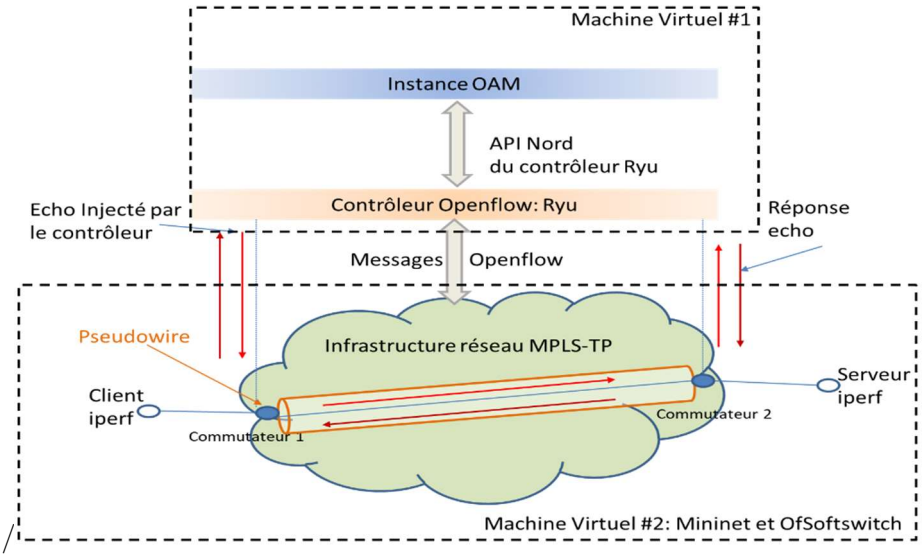


Figure 59 Topologie de référence pour les simulations

b) Résultats des simulations

Nous allons exécuter deux scénarios où à chaque fois on aura différentes valeurs de fréquence d'envoi de probe OAM 200 ms et 500 ms.

Ensuite, nous mesurons le délai à différents débits en utilisant iperf de 1 Mbps à 90 Mbps sachant que l'hôte 1 joue le rôle de client iperf et l'hôte 2 joue le rôle de serveur iperf. Le client iperf envoie des datagrammes UDP de 1470 octets et de taille du tampon UDP fixée par défaut à 208 Ko. Chaque test dure 100 secondes et c'est la valeur moyenne qui est acceptée.

Les valeurs RTT récupérées ont été mesurées pour le commutateur 1 et le commutateur 2 dans les mêmes conditions. La Figure 60 montre la variation du délai par rapport à l'augmentation de trafic qui est exprimé en bande passante. Le nombre de sauts (hops) est aussi un paramètre important et qui est représentatif de la taille du réseau de transport.

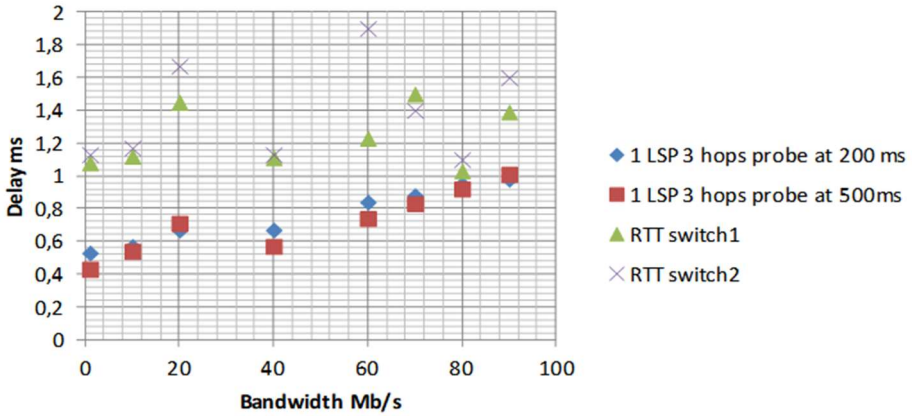


Figure 60 Test #1 : une instance OAM par LSP

Les valeurs RTT obtenu sont proportionnelles à l'augmentation de la bande passante. Ce comportement était prévisible puisque l'augmentation de la bande passante implique systématiquement une augmentation au niveau de la sérialisation qui est l'opération de préparation de paquet en sortie d'une interface.

Dans le cas de test suivant, nous avons créé une topologie linéaire avec trois commutateurs, puis avec cinq commutateurs et enfin avec dix commutateurs. Ces commutateurs sont disposés d'une manière linéaire entre les commutateurs 1 et 2.

Pour chaque topologie, nous avons établi un LSP de bout en bout et avons exécuté une instance OAM pour chaque LSP. Ensuite, nous avons mesuré le délai pour différents débits à l'aide d'iperf, Figure 61.

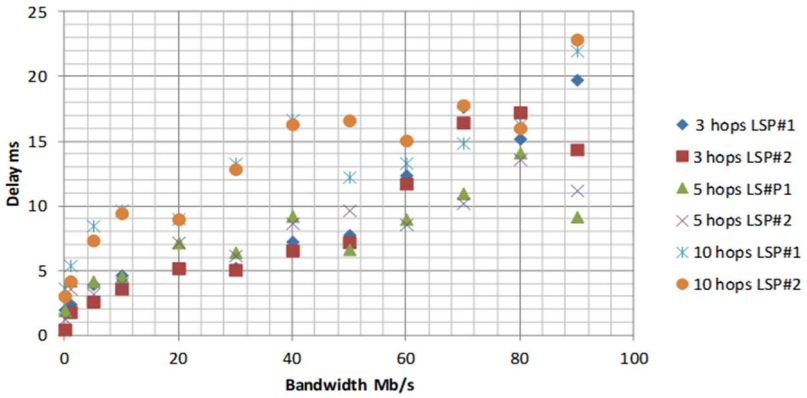


Figure 61 Test #2 : une instance OAM par LSP (nombre de LSP: 2, nombre de hops: 3, 5 et 10)

Les tests confirment que l'augmentation de hop sur un LSP ne fait pas augmenter le délai RTT d'une manière anormal. Ces valeurs de délai sont tout à fait comparables à celles relevées sur un réseau classique MPLS-TP.

Pour ce dernier scénario, nous avons essayé de stresser notre maquette en ajoutant plus de LSP sur une topologie linéaire de dix commutateurs, Figure 62.

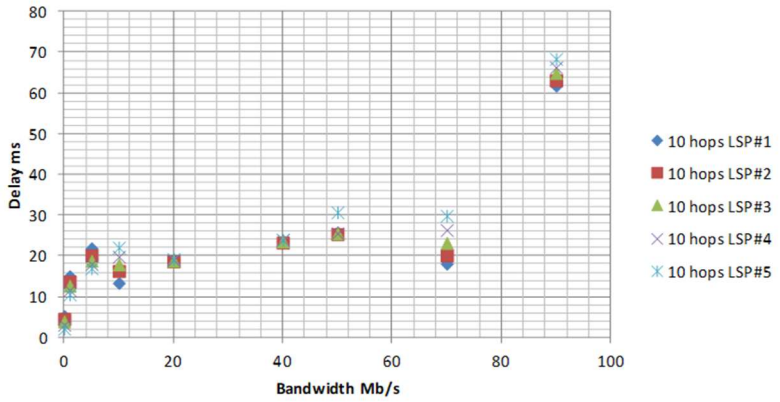


Figure 62 Test #3: une instance OAM par LSP (nombre de LSP : 5, nombre de hops: 10)

Pendant la phase de test #1, l'augmentation de la bande passante, générée à l'intérieur d'un même chemin de données utilisateur, n'affecte pas la variation de délai. La valeur maximale ayant été mesurée ne dépassait pas la valeur de 1,9 ms.

Le deuxième cas de test montre comment les délais augmentent de deux à trois fois lorsque l'on compare le scénario des deux Pseudowires monitorés avec celui des cinq Pseudeuwire monitorés. Le fait d'exécuter simultanément l'instance OAM sur chaque Pseudowire augmente la charge CPU du contrôleur et donc la capacité de traiter l'événement Packet-In qui est la partie du module OAM où le calcul du délai est effectué.

Le dernier scénario de stress confirme le résultat du deuxième scénario puisque le délai augmente de façon spectaculaire avec près de quarante fois le délai calculé dans le premier scénario.

5.4. Conclusion

L'application du paradigme SDN sur un réseau MPLS-TP a montré sa faisabilité tout en respectant les normes sur lesquelles se basent les réseaux MPLS-TP. L'ajout d'autres modules OAM, routages ou autres peut facilement être envisageable sans pour autant mettre à jour les éléments du réseau MPLS-TP. Le service OAM de mesure de délai, que nous avons proposé et testé, a démontré son efficacité. Les valeurs obtenues sur des LSP à travers plusieurs sauts, et sur un réseau MPLS-TP « moyennement chargé » avec plusieurs LSP concurrent ont été satisfaisants. Bien entendu, d'autres méthodes OAM peuvent être exécutées comme application au niveau du contrôleur SDN et cette proposition peut être étendue à un concept plus large qui est l' « OAM-as-a-Service ». Cette facilité, avec laquelle on peut ajouter de nouvelles fonctionnalités, n'est pas du tout évident dans le cas des réseaux de transport MPLS-TP classiques tel qu'on les connait aujourd'hui.

Vue les avantages du SDN, nous présentons dans le chapitre suivant un modèle pour optimiser le routage dans les réseaux MPLS-TP basé sur le paradigme SDN. En effet, nous utilisons un algorithme d'optimisation de routage qui sera basé sur les résultats du plan de gestion en l'occurrence les résultats des outils OAM. On conclura le chapitre avec des simulations et des comparaisons avec d'autres algorithmes d'optimisation de routage classiques.

Chapitre 6

Proposition de solution d'optimisation de routage basée sur SDN et sur les outils OAM

6.1. Introduction

On s'est penché, lors du chapitre précèdent, sur l'adoption du paradigme SDN au niveau des réseaux MPLS-TP, et nous avons pu développer un service basé sur un outil OAM de mesure de délai utilisant l'API nord d'un contrôleur SDN (Ryu). Dans ce chapitre, on développe cette idée pour proposer un concept plus large permettant d'optimiser le routage au niveau des réseaux MPLS-TP en s'appuyant entre autres sur les résultats des outils OAM tout en respectant les contraintes dictées par les SLA. Les SLA étant les engagements sur la qualité de service fourni par un opérateur à son client.

Cette approche permet d'étendre la liste des contraintes adoptées déjà par les protocoles de routage les plus courant comme OSPF et qui sont généralement : la bande passante, le délai, le poids, et le nombre de sauts/liens. En effet, grâce à la flexibilité du paradigme SDN, d'autres contraintes résultantes des outils OAM peuvent être rajouté pour une ingénierie de trafic plus efficace et satisfaire des SLA de plus en plus exigeantes.

Dans la suite de ce chapitre, on décrira la problématique ainsi que la solution proposée.

Les résultats des différentes simulations correspondants à plusieurs algorithmes sont présentés et analysés également à la fin de ce chapitre

6.2. Optimisation du routage dans les réseaux de transports

Le routage, qui est le processus de sélection des chemins dans un réseau le long duquel le trafic réseau doit être envoyé, permet aux réseaux de transport d'assurer la distribution efficace de l'information entre ses utilisateurs. C'est le plan de contrôle qui est responsable d'exécuter le processus de routage.

Un problème fondamental d'optimisation des réseaux MPLS-TP est de savoir comment trouver plusieurs chemins LSP répondant à des SLAs qui sont déclinés sous des contraintes engagées entre le fournisseur de service et son client. Le respect des SLA est une clause très importante qui risque, en cas de non-conformité, de dégrader le service souscrit par le client

et parfois peut être sujet à un paiement de pénalités par l'opérateur aux clients dont les SLAs sont violées.

Comme contraintes, on peut trouver à titre d'exemple:

- La garantie en bande passante
- Le délai de transit réseau
- Le taux de perte de paquet
- La variation de délai
- La disponibilité

Parmi ces contraintes SLA certaines sont mesurables via l'utilisation d'OAM de « Mesures de Performance » comme: le taux de perte, le délai de transit…

Ce type de problème multi contraintes est connu comme un problème NP-complet [80] puisqu'on essaie à la fois de satisfaire les contraintes de SLA mais également d'optimiser les ressources du réseau. L'objectif n'est pas de trouver le meilleur chemin mais de trouver un des chemins qui satisfait les contraintes.

Ceci nous mène à étudier différents algorithmes pour optimiser le routage au sein de notre réseau de transport. Nous proposerons également notre propre modèle permettant d'utiliser les outils OAM afin d'améliorer la qualité de notre chemin optimal.

6.2.1. Algorithme Dijkstra

Dans la recherche du chemin le plus optimal, certains algorithmes se contentent de résoudre le problème du chemin le plus court. Leur objectif est de déterminer le chemin au « moindre coût » entre deux nœuds, où la somme des coûts des bords constitutifs est minimisée.

De nombreux algorithmes de routage existent pour résoudre les chemins les plus courts. Le plus connus d'entre eux est l'algorithme de Dijkstra. L'algorithme de Dijkstra, conçu par l'informaticien néerlandais Edgar Dijkstra en 1959, est un algorithme de recherche de graphes qui résout le problème du chemin le plus court pour un graphe avec des coûts de liens non négatifs, produisant un arbre de chemin plus court [81].

Les protocoles OSPF (Open Short Path First) et IS-IS (Intermediate System to Intermediate System) sont parmi les protocoles de routage les plus utilises et qui se basent sur l'algorithme Dijkstra.

L'algorithme de Dijkstra utilise une structure de données pour stocker et interroger des solutions partielles triées par distance depuis le début. L'algorithme d'origine utilise une file d'attente à priorité minimale et s'exécute en temps : $O(|V|^2)$

Où $|V|$ est le nombre de nœuds puisque le temps d'exécution augmente de façon quadratique en fonction de la taille des éléments à trier.

Les principales étapes de l'algorithme Dijkstra sont:

- Soit s le nœud de départ, et $w(i, j)$ le poids du lien i, j ;
- Créer une matrice de distance $dist$ pour tous les sommets du graph, en supposant que $dist(s)= 0$ et $dist(v)=\infty$ s'il n'y a aucun lien;

- Créer une file d'attente prioritaire Q, où la priorité est une distance du nœud de départ s ;
- Répéter jusqu'à ce que Q n'est plus vide:
    - Retirer de la file d'attente le sommet u ayant la priorité la plus faible,
    - Pour chaque voisin v du sommet u, **dist(v) = min (dist(u) + w(u,v), dist(v))**
- La dernière ligne de la matrice *dist* est un vecteur contenant les valeurs de distance les plus courtes de s à tous les sommets du graphe.

La Figure 63 présente un exemple de déroulement de l'algorithme de Dijkstra.

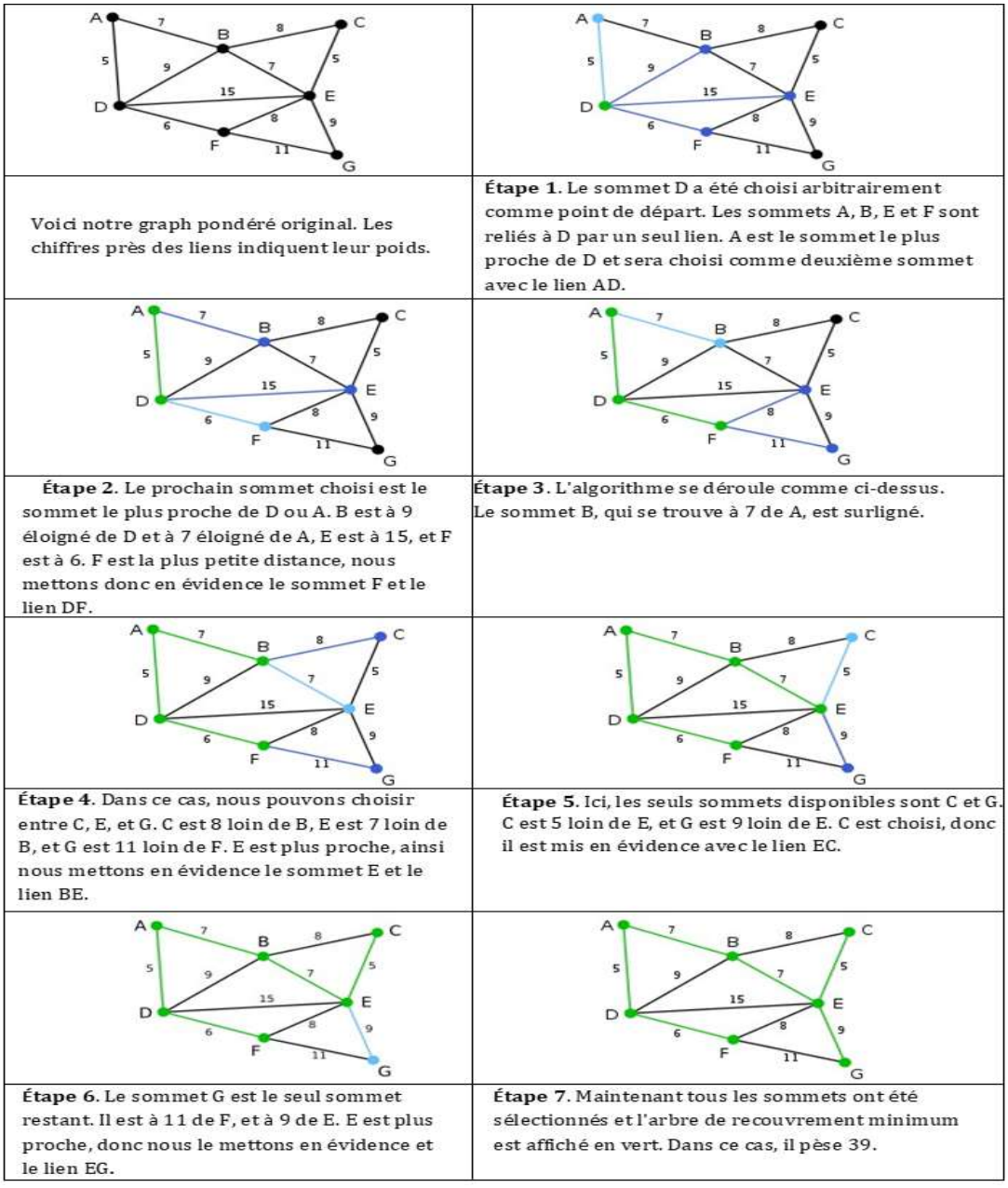


Figure 63 Exemple du déroulement de l'algorithme Dijkstra

6.2.2. Algorithme de Colonie de fourmis

L'observation de la nature est une source d'inspiration pour la recherche de nouvelles solutions. L'une d'entre elles est l'optimisation de la colonie de fourmis qui provient des insectes (fourmis) vivant en colonies. Il a été prouvé que les colonies de fourmis ont des capacités d'optimisation naturelles.

Les algorithmes méta-heuristiques comme l'algorithme d'optimisation des colonies de fourmis ACO (Ant Colony Optimization) peuvent fournir une solution acceptable [82]. Dans les sections suivantes, nous étudions l'un de ces algorithmes qui est basé sur ACO. L'algorithme ACO utilise le même processus que celui utilisé par les fourmis pour joindre leurs aliments, comme le montre la Figure 64.

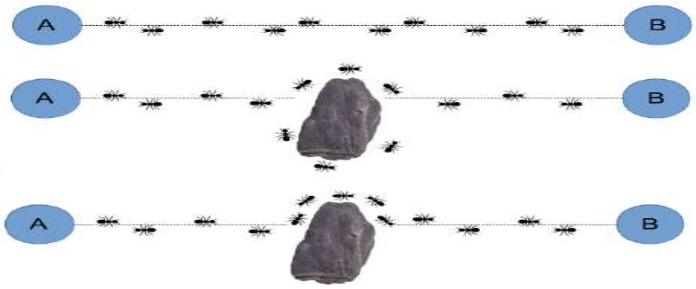


Figure 64 Comportement des fourmis pour rejoindre leur nourriture

1. Les fourmis déposent une matière appelée « phéromone » pour indiquer le chemin qu'elles ont pris du nid vers la nourriture.
2. Les prochaines fourmis choisissent leur chemin en fonction de la concentration de phéromones qu'elles sentent en utilisant un choix probabiliste qui évolue continuellement.
3. Chaque fourmi réitère ce processus, ainsi la quantité de phéromones est mise à jour chaque fois qu'un chemin est choisi. Par conséquent, le chemin le plus court contient une forte concentration de phéromones.

C'est justement cette intelligence collective, basée sur ce concept de dépôt ou d'évaporation de la phéromone par tous les membres de la colonie de fourmis, qui peut être utilisé pour résoudre des problèmes d'optimisation combinatoire ou de satisfaction de contraintes. C'est dans ce cadre-là que nous allons utiliser cet algorithme.

Un réseau MPLS-TP peut être représenté sous la forme d'un graphe non dirigé $G=(V,E)$, où V représente les nœuds et E représente les liens reliant les nœuds. Nous considérons que s'il y a un lien $e=(u,v)$ du nœud $u \in V$ vers le nœud $v \in V$, alors le lien de v vers u existe et que $e=(u,v)=(v,u)$.

L'algorithme ACO indique qu'à chaque période de recherche t, chaque fourmi $k=1,\ldots,m$ choisie aléatoirement un chemin depuis une source i vers la destination j en fonction de la probabilité de ce chemin. La probabilité p qu'une fourmi k se déplace du nœud i vers un nœud j, où ces nœuds appartiennent à l'ensemble des nœuds S_i^k que la fourmi k n'a pas encore visité, est alors:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{a} \times [\eta_{ij}]^{b}}{\sum_{l \in S_i^k} [[\tau_i(l)]^{a} \times [\eta_{il}]^{b}]} \qquad (1)$$

- τ_{ij} est l'intensité du phéromone qui est égale à :
    - τ_0 quand t=0
    - η_{il} quand t≠ 0. η_{il} est une information statistique appelée aussi la "visibilité", elle permet de guider les fourmis vers les nœuds "les plus proches".
- a et b sont des facteurs d'influence: on mettra a à 0 pour que la "visibilité" soit favorisée et on positionnera b à 0 quand les traces de phéromone sont prises en compte lors du déplacement d'un nœud à un autre.

La mise à jour des phéromones est exécutée à chaque itération par une fourmi k le long d'un lien appartenant au chemin selon cette formule :

$$\Delta\tau_{ij}{}^{k}(t) = \frac{Q}{L^{k}(t)} \qquad (2)$$

Où $L^k(t)$ est la longueur du chemin parcouru pendant un cycle par la *kième* fourmi. Et Q est un paramètre constant positif du modèle. Dans notre cas, on fixera Q à 1000 pour que les graphes soient à l'échelle et deviennent plus lisibles.

D'une manière général, la règle pour mettre à jour la quantité de phéromone pour chaque itération est :

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (3)$$

Où $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}(t)$

Où ρ est le coefficient de vaporisation du phéromone et m est le nombre de fourmis qui mettent à jour les informations sur le phéromone.

Le processus de mise à jour du phéromone et de décision de chemin est résumé comme suit:

1. Le cycle démarre avec une quantité de phéromone τ_0 égale par défaut à 0,25. Les autres valeurs sont fixes comme par exemple le cycle de recherche maximal.
2. La formule (1) est utilisée pour calculer la probabilité de choisir le nœud suivant jusqu'à ce que les fourmis trouvent un chemin vers la destination ou que le temps soit dépassé (la valeur *Tmax* est fixée pour limiter le temps de recherche de chemin).
3. Nous recommençons une nouvelle itération (*t+1*) jusqu'à ce que le cycle de recherche maximal soit atteint.
4. Le processus se termine quand un chemin est trouvé.

6.2.3. Modèle amélioré de l'algorithme de Colonie de fourmis : ACO-OAM

En étudiant de plus prêt AnNet qui une variante spéciale de l'algorithme ACO traitant plus particulièrement les problèmes multi-contraintes [83], nous avons pu inclure les

résultats des outils OAM en les positionnant comme contraintes. Cette modification permet d'étendre la notion de contraintes, historiquement limitées au coût et à la bande passante, à d'autres caractéristiques réseaux beaucoup plus fines comme : la gigue, la latence, le temps de transit aller-retour… Cette implémentation est rendue possible grâce au modèle SDN qui est ouvert et qui nous permet d'utiliser l'API nord du contrôleur pour faire exécuter notre code OAM.

La démarche de l'algorithme pour résoudre le problème des multi-contraintes est de faire en sorte que les Nœuds envoient des agents fourmis à leurs voisins selon des intervalles périodiques et de manière aléatoire tout en leur assignant une quantité de phéromones lors de leur passage. La quantité de phéromones est mise à jour lorsqu'un nœud voisin est plus sollicité et augmente la probabilité de prendre ce nœud comme saut suivant dans le LSP.

Une autre modification consiste à rendre l'algorithme approprié à la technologie MPLS-TP. En effet, on doit se soumettre à certaines exigences liées au réseau de transport comme ne pas permettre le multi-trajet (pas de ECMP) et de toujours prendre le même chemin pour les deux sens aller et retour.

On considère un chemin $p=p(s,d)$ qui représente le trajet du nœud source s vers le nœud destination d. On dira que chaque chemin p est caractérisé par plusieurs paramètres QoS: le délai $D(p)$, la largeur de bande $B(p)$, et le taux de perte de paquet $PL(p)$.

Par souci de simplification, nous supposons que, pour chaque liaison e, les paramètres de qualité de service sont égaux dans le sens retour et dans le sens aller de v vers u et de u vers v.

Le délai du chemin p désigne les délais concaténés causés par les opérations de traitement, de transmission, de propagation et de la mise en file en attente. Il est donc égal à la somme des délais de toutes ses liaisons:

$$\boldsymbol{D\big(p(s,d)\big) = \sum_{e\in p(s,d)} D(e), \quad e \in p(s,d)} \qquad (4)$$

La largeur de bande passante disponible du chemin p est le minimum de bande passante de toutes ses liaisons:

$$\boldsymbol{B\big(p(s,d)\big) = Min\{B(e), e \in p(s,d)\}} \qquad (5)$$

L'objectif est de trouver un chemin p ou un ensemble de chemins (chemin de primaire et chemin de back-up) qui répondent aux contraintes de délai et de bande passante où :

Le délai total du trajet ne doit pas dépasser *Dmax* :

$$\boldsymbol{D\big(p(s,d)\big) \le D_{max}} \qquad (6)$$

La bande passante minimale doit être inférieure à *Bmin* :

$$\boldsymbol{B\big(p(s,d)\big) \ge B_{min}} \qquad (7)$$

Et le taux de perte de paquet (Packet Loss) ne doit pas dépasser *PLmax*:

$$\boldsymbol{PL\big(p(s,d)\big) \le PL_{max}} \qquad (8)$$

Afin de pouvoir évaluer les solutions trouvées par l'algorithme ACO on utilise une fonction fitness $f(e)$ (ou fonction objective : la liste des objets étant le délai, la bande passante et le

taux de perte de paquets) pour évaluer la qualité d'un chemin. Nous définirons la fonction fitness comme suit:

$$\boldsymbol{f(e)} = \begin{cases} \mathbf{0} & \boldsymbol{si} \begin{bmatrix} \boldsymbol{(B(e) < B_{min}) ou\ si\ (D(e) > D_{max})} \\ \boldsymbol{ou\ si\ (PL(e) > PL_{max})} \end{bmatrix}, \\ \boldsymbol{w1 \times e^{-D(e)} + w2 \times e^{-B(e)} + w3 \times e^{-PL(e)}} & \boldsymbol{pour\ les\ autres\ cas} \end{cases}$$

(9)

Où *w1, w2* et *w3* sont des coefficients d'influence grâce auxquels on peut favoriser un critère par rapport à un autre moyennant une logique de pondération. Les critères considérés dans notre étude sont : le délai, la bande passante et le taux de perte de paquets.

La règle suivante s'applique sur les coefficients d'influence:

$w1 + w2 + w3 = 1$ où $w1, w2, w3 \in [0,1]$

Nous pouvons fixer les poids correspondants en fonction des besoins des utilisateurs. Ainsi, pour le reste de ce chapitre les coefficients *w1, w2* et *w3* recevront respectivement une valeur de *0,7, 0,2* et *0,1*. Dans notre cas, ceci implique que le délai sera le facteur prépondérant afin de démontrer la valeur ajoutée des OAM dans les prises de décision de routage.

Ainsi, la fitness *f(p)* du chemin p sera calculée comme suit :

$$\boldsymbol{f(p) = 0.7 \times e^{-D(p)} + 0.2 \times e^{-B(p)} + 0.1 \times e^{-PL(p)}} \qquad (\ 10\ )$$

L'évaluation de la fonction fitness *f(p)* permettra de savoir si un chemin p est éligible selon les critères SLA que nous avons défini au préalable. Plus la valeur de *f(p)* est grande mieux c'est. Dans le cas contraire, la mise à jour du phéromone sur ce chemin doit être en fonction de la valeur de fitness trouvée.

La nouvelle règle pour mettre à jour la quantité de phéromone pour chaque itération est :

$$\boldsymbol{\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t)} \qquad (\ 11\ )$$

Où $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta{\tau_{ij}}^{k}(t)$

$$\boldsymbol{\Delta{\tau_{ij}}^{k}(t) =}$$

$$\begin{cases} \boldsymbol{Qf(k)} & \textbf{quand la kiéme fourmi passe par le lien (i, j) lors du cycle ;} \\ \mathbf{0} & \textbf{pour les autres cas} \end{cases}$$

111

Où :

ρ est le coefficient de vaporisation du phéromone, $\rho \in]0,1[$

$\Delta\tau_{ij}(t)$ représente l'incrémentation de la quantité de phéromone sur le lien *(i,j)* à chaque cycle.

m est le nombre de fourmis qui mettent à jour les informations sur le phéromone.

Q est un paramètre constant positif du modèle. Dans notre cas, on fixera Q à 100.

Et *f(k)* représente la fitness du chemin parcouru par la kième fourmi lors du cycle

Le processus de mise à jour de la phéromone et de décision de chemin de l'algorithme ACO-OAM se distingue par le fait qu'une solution de chemin n'est retenue que si elle est conforme aux SLA exigées. Les principales étapes de l'algorithme proposé sont comme suit :

1. Le cycle démarre avec une quantité de phéromone τ_0 égale par défaut à 0,25. La fonction Fitness est initialisée à 0, les autres valeurs sont fixes comme par exemple le cycle de recherche maximal.
2. La formule (1) est utilisée pour calculer la probabilité de choisir le nœud suivant jusqu'à ce que les fourmis trouvent un chemin vers la destination ou que le temps soit dépassé (la valeur *Tmax* est fixée pour limiter le temps de recherche de chemin).
3. Une fois un chemin trouvé, les fonctions de mesure de SLA (dans ce cas précis : délai et taux de perte de paquets, qui font partie des outils OAM), sont exécutées sur ce chemin de bout en bout [84].
4. On calcule après la fonction fitness à l'aide de la formule (10) et on sélectionne le chemin avec la valeur maximale de f(e) tout en mettant à jour la phéromone.
5. Nous recommençons une nouvelle itération (*t+1*) et revenons à l'étape 5 jusqu'à ce que le cycle de recherche maximal soit atteint.
6. On sort du cycle quand un chemin répondant aux exigences de délais demandés. Dans le cas échéant on choisit le chemin ayant la meilleure valeur de fitness.
7. Ensuite, nous continuons nos mesures SLA sur le LSP sélectionné à intervalle régulier [85]. Si une violation est constatée et qui dure dans le temps (intervalle paramétrable), on reprend le cycle depuis la phase 1 jusqu'à la phase 6. Le nouveau LSP est ainsi poussé depuis le contrôleur SDN vers les commutateurs SDN afin de re-router le trafic.

La Figure 65 illustre la charte du déroulement de l'algorithme ACO-OAM.

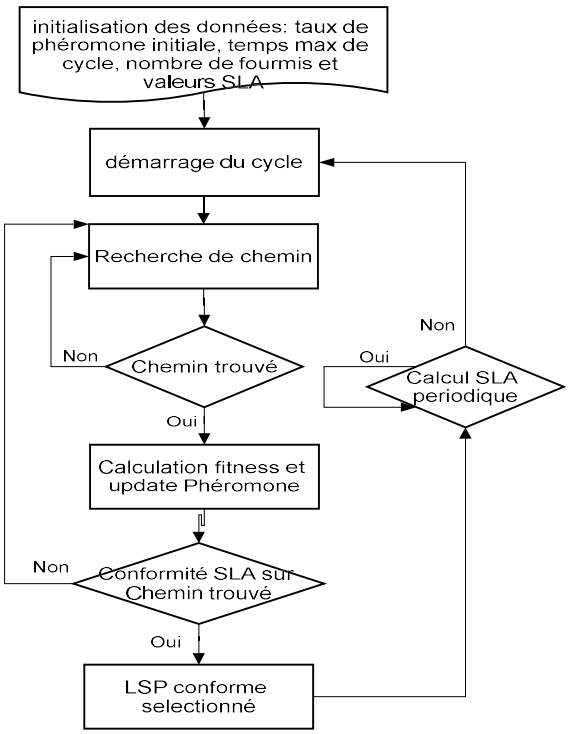


Figure 65 Charte de l'algorithme ACO-OAM

6.3. Tests et Résultats des simulations

Ce chapitre présente les expériences qui ont été menées afin d'évaluer l'efficacité des différents modèles et algorithmes présentés et proposés (Dijkstra, Colonie de fourmis standard, et Colonie de fourmis modifié). Ces expériences visent à tester les différents modèles et les algorithmes qu'ils contiennent, ainsi qu'à prendre en compte d'autres propositions. Les résultats obtenus avec les différents algorithmes sont comparés.

6.3.1. Environnement des expérimentations

Deux machines sont utilisées pour les essais expérimentaux. La première machine dispose d'un processeur quadruple cœur Intel i5 2,5 GHz sur laquelle tourne notre contrôleur SDN Ryu ainsi que les différents codes applicatifs écrit en Python: Découverte de topologie, collecteur de statistiques pour la bande passante, fonction de mesure de délai et surtout la fonction principale de calcul de chemin liées aux différents algorithmes Dijkstra, ACO et ACO-OAM. La seconde machine possède un processeur Intel huit cœurs 3,7 GHz permettant d'exécuter Mininet avec plusieurs Openvswitch, et d'émuler des hôtes qui permettent de générer différents types de flux via Iperf [86].

Le banc d'essai est bien illustré dans la Figure 66:

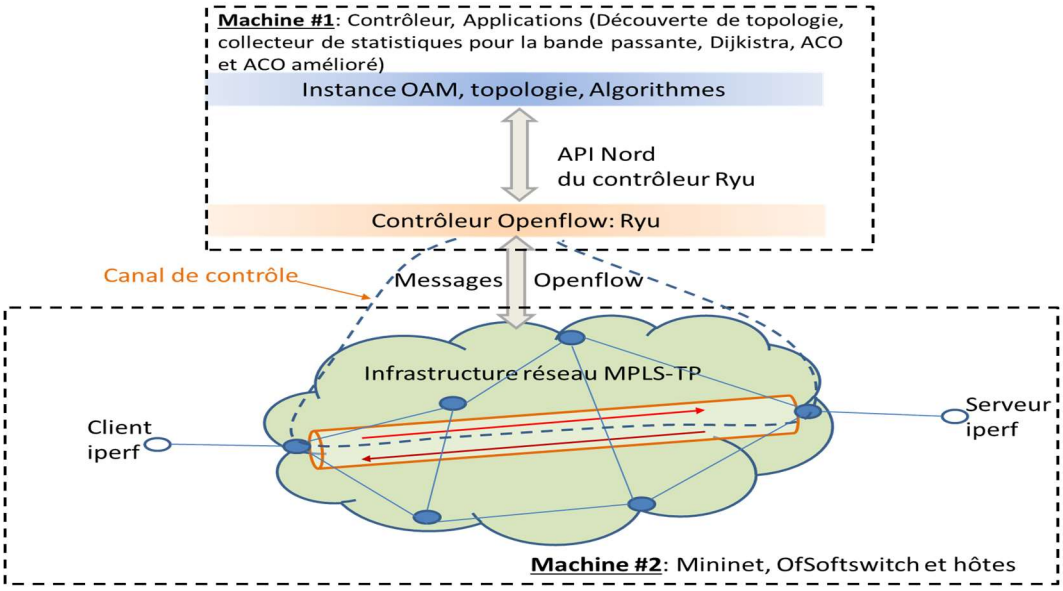


Figure 66 Architecture logique de notre banc d'essai

Les applications « dijkstra_ryu.py », « aco_ryu.py » et « aco_oam_ryu.py » sont écrites en Python et utilisent l'API nord du contrôleur Ryu. Elles disposent de plusieurs fonctions parmi lesquelles on peut trouver notamment les fonctions de collecte de statistiques, de découverte de topologie, et de recherche et de programmation de chemin. On notera la réutilisation du module OAM, décrit dans la section 5.3.2.

Différentes topologies sont lancées afin de vérifier le comportement et mesurer la performance des trois algorithmes avec un nombre de nœuds de 10, 20, 30, 40 et 50, chacun ayant respectivement 20, 30, 62, 70 et 80 liens, Figure 67.

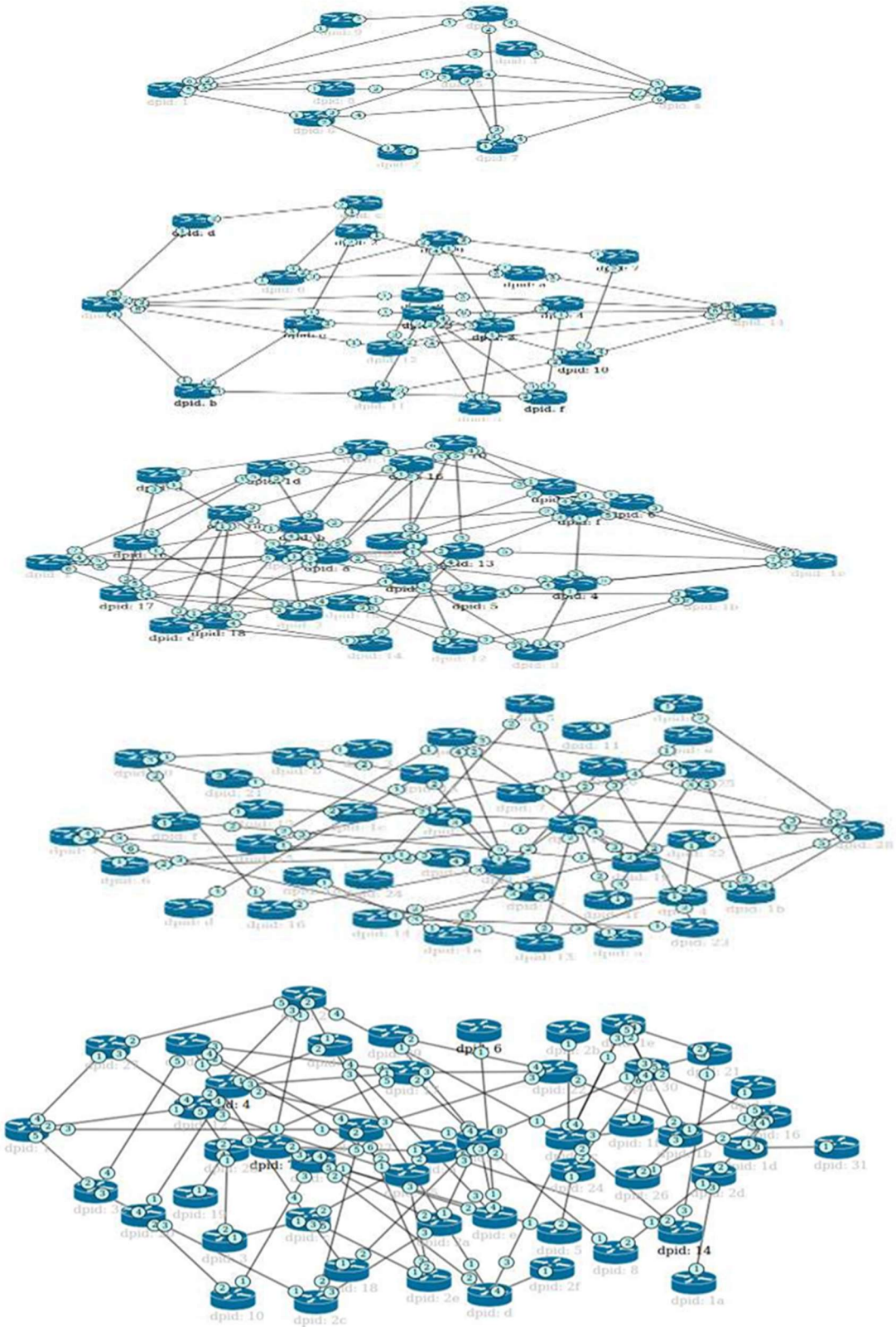


Figure 67 Topologies utilisées pendant les différents essais

6.3.2. Méthodologie des tests et des critères d'évaluations

Dans cette section, on expose la démarche des simulations ainsi que les critères d'évaluations. On termine par l'analyse des résultats expérimentaux.

a) Méthodologie des tests

Tous les tests sont effectués sur les 5 topologies (Tableau 7) et chaque test unitaire est répété 5 fois d'une manière indépendante et pour chacune des expérimentations. La durée de chaque expérimentation est fixée à 120 secondes.

Topologie	Nombre de Nœuds	Nombre de liens
1	10	20
2	20	30
3	30	62
4	40	70
5	50	80

Tableau 7 Les topologies utilisées pendant les simulations

Il est aussi important de noter qu'il y a un nombre de paramètres que nous avons fixé afin d'optimiser l'exécution des algorithmes ACO et ACO-OAM. Parmi lesquels on peut citer :

- Le nombre maximal de nombre de sauts sur un chemin est limité à 15. Toute solution au-delà de 15 sauts est considérée en violation avec les SLA et donc pas retenue.
- Le temps maximum d'exécution au bout duquel on arrête le cycle est fixé à 5 secondes. Au-delà de cette valeur, on considère que l'algorithme a échoué.
- Le nombre total m de fourmis est fixé à 40 et le coefficient de vaporisation de la phéromone ρ est réglé à 0,5.
- Le LSP est toujours calculé depuis la source qui est toujours le premier nœud (Node-id=1) et la destination qui est le dernier nœud (Node-id le plus grand) dans la topologie à tester.

b) Les critères d'évaluation

Afin d'évaluer les performances des différents algorithmes, nous avons choisi les critères suivants:

- Le nombre total de sauts pour chaque chemin.
- Temps d'exécution de chaque algorithme jusqu'à convergence. Ceci représente un coût en terme de ressources CPU mais également mesure l'efficacité de chaque algorithme.

- Le comportement et la réactivité de chaque algorithme vis-à-vis d'une violation de SLA. Deux mesures sont à prendre en compte : la perte de paquet et le délai aller-retour de bout-en-bout.
- Un dernier élément aussi important est celui du taux de succès qui constitue la capacité de chaque algorithme à converger dans un temps « raisonnable » (le temps maximum déjà cite dans le paragraphe précèdent)

6.3.3. Analyse des résultats et conclusions

Dans cette section nous exposons les résultats de plusieurs simulations qui ont été réalisées afin de comparer les performances de chaque algorithme. Les caractéristiques liées aux flux générés et à la bande passante des liens. Aucun changement n'est effectué au cours de la simulation sauf pour le dernier test où on aura à faire varier le débit généré suivant les valeurs 10 Mbps, 40 Mbps et 70 Mbps.

La Figure 68 montre que l'algorithme dijkstra est meilleur en terme de temps de convergence et donc moins consommateur en ressource CPU. Ceci s'explique par le fait qu'il se base uniquement sur les coûts de chaque liaison du réseau. L'algorithme ACO est mieux que ACO-OAM en termes de temps de convergence puisqu'il a une partie de moins dans ces blocs de traitement, et plus précisément de la phase de calcul de la fonction fitness.

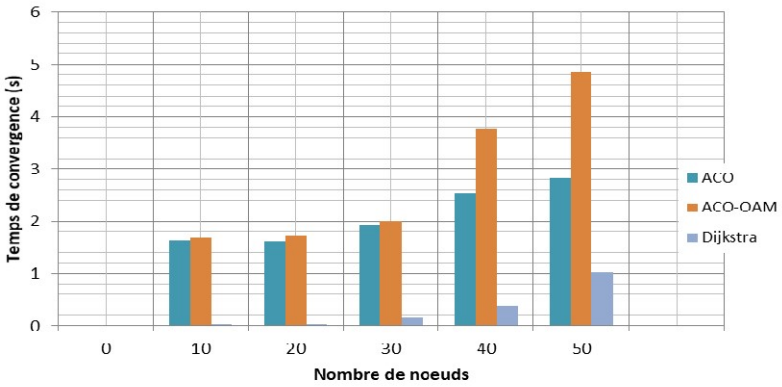


Figure 68 Temps de convergence par algorithme

D'un point de vue nombre de sauts, la Figure 69 montre aussi que Dijkstra est mieux que les deux autres algorithmes en terme de nombre de sauts. En effet, il est vrai que les algorithmes basés sur ACO explorent plus de chemins que Dijkstra, ceci est dû essentiellement au nombre de fourmis « m » chargées d'explorer les chemins, plus le nombre m est grand plus on se rapproche des résultats retrouvés par Dijkstra. Le paramètre « m » influe directement sur le temps de convergence de l'algorithme et aussi le nombre de saut pour le chemin sélectionné.

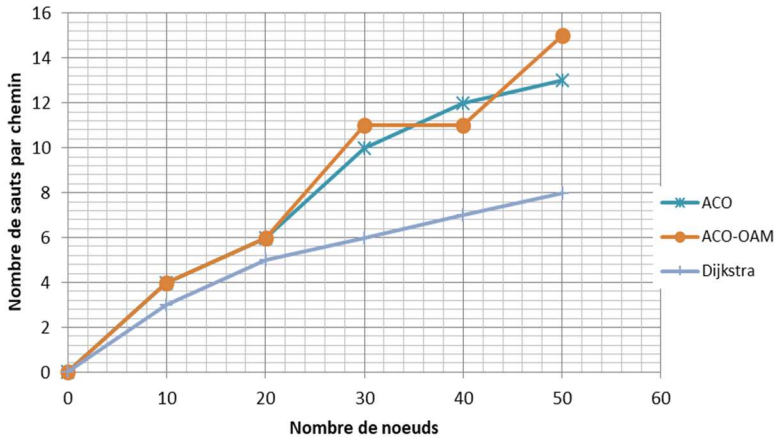


Figure 69 Nombre de sauts par chemin

Les figures suivantes, Figure 70, Figure 71 et Figure 72 représentent la qualité des LSP retrouvé par les trois algorithmes dans des conditions de débits différentes. Ces résultats montrent clairement que le point commun entre ces simulations est que l'algorithme ACO-OAM réagit mieux dans des conditions de stress de SLA. En effet, le module OAM permet de calculer la fitness pour chaque chemin retrouvé et le confronter aux valeurs SLA exigées pour ce LSP.

Les chemins sélectionnés par Dijkstra et ACO présentent des valeurs plus élevées en terme de délai de bout en bout. Ceci est dû au fait que CO-OAM mesure le délai de bout en bout via des OAM, alors que ACO et Dijkstra mesure le délai pour chaque lien à part.

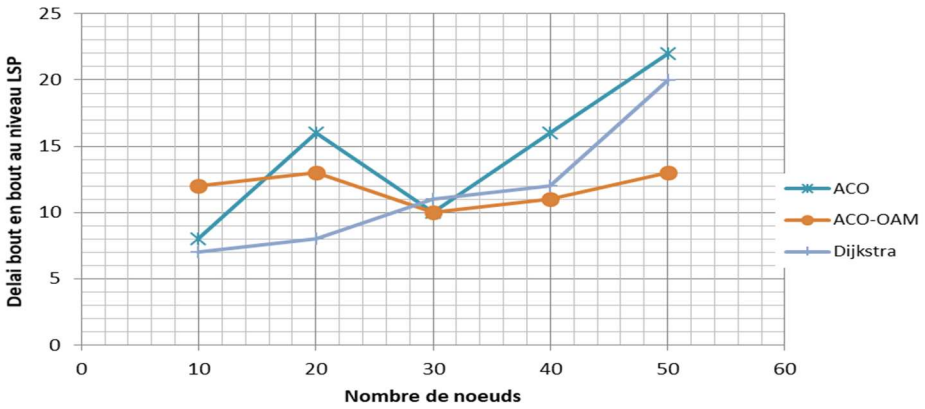


Figure 70 Délai de bout en bout sur un LSP en présence de contraintes SLA: débit à 10 Mbps

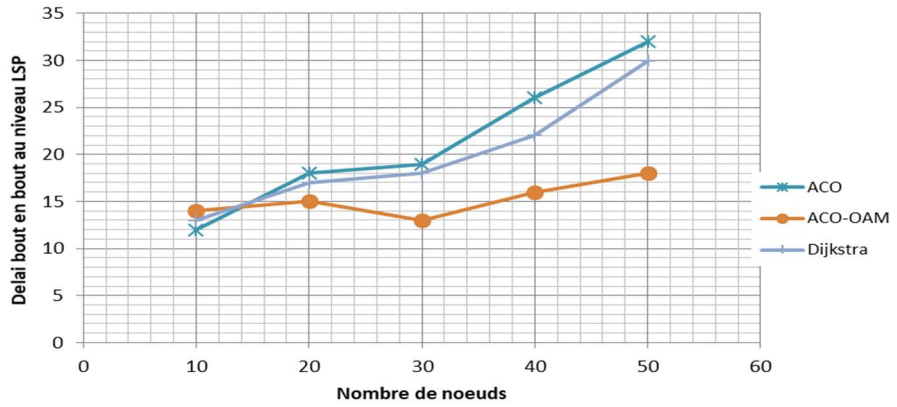


Figure 71 Délai de bout en bout sur un LSP en présence de contraintes SLA: débit à 40 Mbps

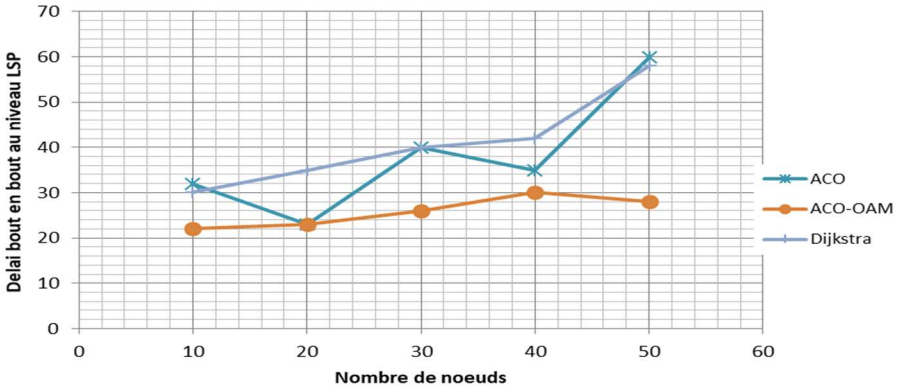


Figure 72 Délai de bout en bout sur un LSP en présence de contraintes SLA: débit à 70 Mbps

Compte tenu des exigences SLA, on ne considère un chemin accepté que si son délai calculé de bout en bout est conforme. La Figure 73 illustre bien cette hypothèse puisque ACO-OAM réalise un taux de succès mieux que les deux autres algorithmes. Le taux de 100% n'a pas été atteint pour ACO-OAM, puisque pour certains cas il a fallu augmenter le Temp maximal de recherche et le nombre de fourmis dans la colonie.

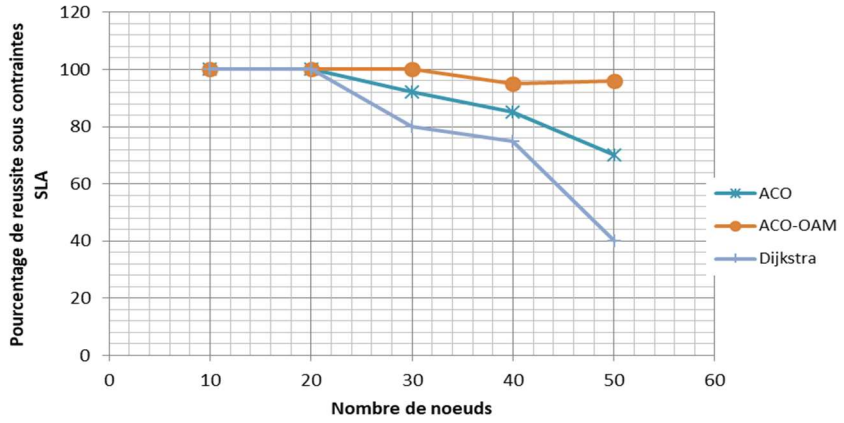


Figure 73 Taux de succès de chaque algorithme en cas de présence de SLA

6.4. Conclusion

Il ressort clairement des simulations qui ont été conduites sur les différents algorithmes que Dijkstra montre une meilleure performance en termes de temps de convergence et de nombre de sauts. Par rapport à ce critère-là, les algorithmes ACO ne font pas mieux que Dijkstra.

Par contre, on constate que Dijkstra, étant dépendant du trafic (bande passante) et de la topologie, est moins sensible au délai. ACO réussit à explorer l'ensemble du réseau ce qui donne, dans l'ensemble, un trafic mieux distribué dans le réseau et par conséquent moins exposé à des problèmes de violation de SLA.

Les simulations ont démontré que l'algorithme ACO-OAM proposé est plus lent mais il est « sûr ». En effet, en plus du fait qu'il explore le réseau comme ACO, il permet de ne sélectionner que les LSP conformes aux exigences SLA. Avec l'augmentation de puissance des unités de calcul, cette lenteur ne sera plus un handicap.

Chapitre 7

Conclusion générale

L'aire des réseaux SDH/PDH étant révolue, le monde du transport s'est penché sur des technologies évolutives et moins coûteuses. Parmi ces nouvelles technologies de transport, nous avons fait le choix d'étudier MPLS-TP. Une technologie qui a hérité à la fois des standard IP/MPLS mais également des réseaux de SDH/PDH notamment pour leurs avancées en termes de temps de restauration ou de protection de circuit.

Au cours de cette thèse, nous nous sommes intéressés à étudier les technologies de transport nouvelle génération basées sur Ethernet. Nous nous sommes penchés à étudier, plus particulièrement, les différents standards liés aux techniques OAM au sein des réseaux de transport basé sur MPLS-TP. Après avoir soulevé les problématiques des outils OAM dans le monde MPLS-TP et fait l'état de l'art des différents standard OAM, nous avons proposé des solutions pour résoudre les problèmes d'interopérabilité et d'interfonctionnement ([87], [6]. Plusieurs simulations ont été élaborées afin de tester notre proposition.

Nous avons également proposé un mécanisme d'auto-découverte de standard OAM, et qui est également une variante du modèle de cloisonnement [7]. Ce mécanisme n'altère pas les standards OAM mais le contourne en offrant aux différents éléments du réseau la possibilité de négocier une même famille d'OAM.

Par la suite, nous avons appliqué le paradigme SDN sur les réseaux MPLS-TP et nous avons proposé une solution pour lancer ces OAM comme « OAM-as-a-Service », [8]. Ceci a été possible grâce au paradigme SDN qui offre la possibilité de programmer le réseau sans pour autant modifier les standards plus particulièrement les plans de données et de contrôle. En effet, nous démontrerons que grâce au SDN on arrivera à la fois à résoudre les problèmes d'interopérabilité et d'Interfonctionnement des OAM, et aussi à donner un nouveau sens à l'utilisation même des OAM en matière d'ingénierie de trafic, d'amélioration des performances et d'optimisation de ressources.

Dans le même esprit, nous avons étendu ce concept de « OAM as a service » à un autre domaine qui est celui de l'optimisation du routage. Bien qu'initialement, les OAM soient plutôt du domaine de plan de gestion, nous avons pu les intégrer dans notre contribution visant à les utiliser aussi dans le plan de contrôle. En effet, on s'est intéressé à étendre l'utilisation des OAM dans les prises de décision du plan de contrôle jusque-là disjoint complétement du plan de gestion, [9].

Cette dernière idée nous a ouvert plusieurs perspectives. Nous avons alors ciblé plus précisément une problématique des opérateurs de transport qui est celle de respecter les SLA contractuels avec leurs clients. En effet, afin de pouvoir respecter leur SLA, les opérateurs de transport sont souvent obligés de sur-dimensionner leur infrastructure pour éviter la saturation de leur réseau. Ce qui impacterait d'une manière directe leur budget d'infrastructure et par

conséquent les coûts des services rendu. Nous avons alors fait le choix d'un algorithme d'optimisation basé sur les colonies de fourmis et on l'a doté de la puissance des OAM grâce au SDN afin de délivrer un routage optimisé et respectant les SLA [10].

En terme de perspective, nous estimons qu'un autre point serait intéressant pour la suite de ce travail c'est de reprendre l'algorithme modifié de colonie de fourmis et d'y inclure les recherches de chemins de secours pour augmenter la résilience en cas de violation de SLA ou tout simplement en cas de changement de topologie du réseau.

On peut également ajouter que le paradigme SDN a mis la barre très haute puisqu'il nous ouvre la possibilité de créer de nouvelles applications réseau autrefois difficile à implémenter. Ainsi, le concept d'optimiser le routage, en se basant sur des algorithmes comme celui de colonie de fourmis, peut être étendu à un concept plus large qu'on nommera le « SLA Based Routing ».

D'autres solutions « vertes » peuvent influer les décisions de routage peuvent être imaginé à travers le prisme SDN comme de la gestion de l'énergie dans les centres de données qui sont souvent très soucieux de leurs factures d'électricité.

Bibliographie

[1] K. Mitra and A. Zaslavsky, "QoE Modelling , Measurement and Prediction : A," pp. 1–25, 2014.

[2] G. Pujolle, O. Salvatori Collaborateur, and J. C. Nozick, *Les réseaux*. Paris: Eyrolles, ISBN: 978-2-212-11757-8, 2008.

[3] L. Martini and D. Engineer, "The New Generation of Transport Networks . NGN transport - Moving from SONET / SDH TDM," pp. 1–23, draft-ietf-PWE3-control-protocol-17.txt, IETF, 2009.

[4] IEEE802.1Qay, "IEEE Standard for Local and metropolitan area networks-Virtual Bridged Local Area Networks Amendment 10: Provider Backbone Bridge Traffic Engineering." pp. c1-131, 2009.

[5] S. Bryant and L. Andersson, *Joint Working Team (JWT) Report on MPLS Architectural Considerations for a Transport Profile*, no. 5317. IETF, 2009.

[6] M. Azizi, R. Benaini, and M. Ben Mamoun, "Key requirements for interworking between MPLS-TP network and IP/MPLS network," *Int. J. Eng. Technol.*, vol. 5, no. 4, pp. 3351–3358, 2013.

[7] M. Azizi, R. Benaini, and M. B. Mamoun, *MPLS-TP: OAM discovery mechanism*, vol. 7593 LNCS. 2013.

[8] M. Azizi, R. Benaini, and M. Ben Mamoun, "The Programmable Cloud Network: Delay Measurement Application," in *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, 2014, pp. 687–693.

[9] M. Azizi, R. Benaini, and M. Ben Mamoun, "Delay Measurement in Openflow-Enabled MPLS-TP Network," vol. 9, no. 3, pp. 90–101, 2015.

[10] M. Azizi, R. Benaini, and M. Ben Mamoun, *Delay-Bandwidth Optimization Method Based on Ant Colony Algorithm Applied to Transport Network Using SDN Paradigm*, vol. 11557 LNCS. Springer International Publishing, 2019.

[11] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco VNI Global Complete Forecast Update," no. December, pp. 2017–2022, 2018.

[12] Infonetics Research, "Microwave Equipement: Quarterly Market Share, Size and Forcast," 2014.

[13] MEF, "Ethernet Services Definitions - Phase 2," *MEF Tech. Specif. 6.1*, 2008.

[14] A. Kirstädter, C. Gruber, J. Riedl, and T. Bauschert, "Carrier-grade ethernet for core networks," *OFC/NFOEC 2007 - Opt. Fiber Commun. Natl. Fiber Opt. Eng. Conf. 2007*, 2007.

[15] IEEE std 802.1ad, "Provider Bridges," 2006.

[16] IEEE Std 802.1ah, "Provider Backbone Bridges," 2008.

[17] IEEE Std 802.1Qay, "Provider Backbone Bridge Traffic Engineering," 2009.

[18] E. Menachi and R. Giladi, "Hierarchical ethernet transport network architecture for backhaul

cellular networks," *Wirel. Networks*, vol. 19, no. 8, pp. 1933–1943, 2013.

[19] IEEE Std. 802.1Q, "Local and Metropolitan Area Networks, Virtual Bridged Local Area Networks," 2003.

[20] IETF RFC 3945, "RFC 3945 Generalized Multi-Protocol Label Switching (GMPLS) Architecture," 2004.

[21] D. Fedyk, H. Shah, N. Bitar, and A. Takacs, "RFC6060: Generalized Multiprotocol Label Switching (GMPLS) Control of Ethernet Provider Backbone Traffic Engineering (PBB-TE)," 2011.

[22] ITU-T G.8110.1, "G.8110.1 Architecture of Transport MPLS (T-MPLS) Layer Network," 2006.

[23] IETF RFC 3031, "RFC 3031: Multiprotocol Label Switching Architecture," *IETF*, 2001.

[24] D. Graham and D. Rockmore, "The packet switching brain," *J. Cogn. Neurosci.*, vol. 23, no. 2, pp. 267–269, 2011.

[25] IETF RFC 4364, "BGP/MPLS IP Virtual Private Networks (VPNs)," *IETF*, 2006.

[26] IETF RFC 4448, "Encapsulation Methods for Transport of Ethernet over MPLS Networks," *IETF*, 2006.

[27] IETF RFC 4447, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)," *IETF*, 2006.

[28] ITU, "P.800: Methods for subjective determination of transmission quality," *ITU-T Recomm.*, vol. 800, 1996.

[29] IETF RFC 1633, "Integrated Services in the Internet Architecture: an Overview," *IETF*, 1994.

[30] IETF RFC 2475, "An Architecture for Differentiated Services," *IETF*, 1998.

[31] "Y.1711 : Operation & Maintenance mechanism for MPLS networks," 2004. [Online]. Available: https://www.itu.int/rec/T-REC-Y.1711-200402-I/en. [Accessed: 07-Aug-2018].

[32] IETF RFC 5880, "RFC 5880: Bidirectional Forwarding Detection," *IETF*, pp. 1–49, 2010.

[33] IETF RFC 8029, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures," *IETF*, 2017.

[34] IETF RFC 8077, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)," *IETF*, 2017.

[35] IETF RFC 4385, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN," *IETF*, 2006.

[36] IETF RFC 5654, "RFC 5654: Requirements of an MPLS Transport Profile," 2009.

[37] IETF RFC 5921, "RFC 5921 A Framework for MPLS in Transport Networks," 2010.

[38] IETF RFC 5960, "RFC 5960 MPLS Transport Profile Data Plane Architecture," 2010.

[39] IETF RFC 6373, "RFC 6373 MPLS Transport Profile (MPLS-TP) Control Plane Framework," 2011.

[40] IETF RFC 5036, “RFC 5036 LDP Specification,” 2007.

[41] IETF RFC 6378, “RFC 6378 MPLS Transport Profile (MPLS-TP) Linear Protection,” 2011.

[42] IETF RFC 5586, “RFC 5586: MPLS Generic Associated Channel,” 2009.

[43] IETF RFC 6371, “RFC 6371 Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks,” 2011.

[44] IEEE Std 802.1ag, “Local and Metropolitan Area Networks, Virtual Bridged Local Area Networks, Amendment 5: Connectivity Fault Management,” 2007.

[45] ITU-T Recommendation Y.1731, “OAM functions and mechanisms for Ethernet based networks,” 2008.

[46] IEEE Std 802.3ah, “Amendment to IEEE Std 802.3 – 2002: Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,” 2004.

[47] MEF16, “Ethernet Local Management Interface,” 2006.

[48] S. Bradner and J. MCQuaid, “RFC2544: Benchmarking Methodology for Network Interconnect Devices,” 1999.

[49] IEEE 802.3ah, “IEEE standard for Ethernet in the First Mile,” 2004.

[50] MEF, “Technical Specification MEF 16 Ethernet Local Management Interface (E-LMI) January 2006,” *Metro*, no. January, 2006.

[51] IETF RFC 6424, “RFC 6426 MPLS On-Demand Connectivity Verification and Route Tracing,” 2011.

[52] IETF RFC 6428, “RFC 6428 Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile,” 2011.

[53] T. Takeda, I. Inoue, and K. Shiomoto, “Thin layer for seamless interconnection of multi-technology transport networks,” *2010 14th Conf. Opt. Netw. Des. Model.*, pp. 1–6, 2010.

[54] K. S. Sobana and R. A. K. Kavitha, “Performance Analysis of SDH Network Management System using DWDM Technique,” *Int. J. Eng. Technol. Sci. Res.*, vol. 2, no. January 2015, 2016.

[55] Z. DING and P. SALTSIDIS, “Interworking between Ethernet and MPLS,” Jan. 2012.

[56] A. Farrel, J.-P. Vasseur, and J. Ash, “A Path Computation Element (PCE)-Based Architecture,” Aug. 2006.

[57] J. L. R. JP. Vasseur, “Path Computation Element (PCE) Communication Protocol (PCEP),” 2009.

[58] O.-P. Lamminen, M. Luoma, J. Nousiainen, and T. Taira, “Control Plane for Carrier-Grade Ethernet Network,” *2009 IEEE Globecom Work.*, pp. 1–6, Nov. 2009.

[59] V. Lopez, B. Huiszoon, J. Fernandez-Palacios, O. Gonzalez de Dios, and J. Aracil, “Path computation element in telecom networks: Recent developments and standardization activities,” in *2010 14th Conference on Optical Network Design and Modeling (ONDM)*, 2010, pp. 1–6.

[60] J. Han, D. Watson, and F. Jahanian, "Topology aware overlay networks," in *Proceedings - IEEE INFOCOM*, 2005, vol. 4, pp. 2554–2565.

[61] M. Bocci and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge," *IETF*, 2009.

[62] A. Ayyangar, "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)," *IETF*, 2008.

[63] "eve-ng," 2016. [Online]. Available: http://www.eve-ng.net/downloads. [Accessed: 10-Dec-2018].

[64] Nokia, "7750 Service Router | Nokia Networks," 2016. [Online]. Available: https://networks.nokia.com/products/7750-service-router. [Accessed: 10-Dec-2018].

[65] M. Azizi, R. Benaini, and M. Mamoun, "MPLS-TP: OAM Discovery Mechanism," *Adv. Infocomm Technol.*, no. July, pp. 1–25, 2013.

[66] ONF, "Open Networking Fundation," 2012. [Online]. Available: https://www.opennetworking.org/.

[67] T. Benson, A. Akella, D. Maltz, I. ser NSDI'09, and Berkeley, "Unraveling the complexity of network management," in *in {Proceedings} of the 6th {USENIX} {Symposium} on {Networked} {Systems} {Design}*, 2009, pp. 335–348.

[68] H. Kim and N. Feamster, "Improving network management with software defined networking," *Commun. Mag. IEEE*, vol. 51, pp. 114–119, 2013.

[69] T. Koponen *et al.*, "Onix: a distributed control platform for large-scale production networks, "in Proceedings of the 9th USENIX conference on Operating systems design," *USENIX Assoc.*, vol. 2010, pp. 1–6, 2010.

[70] K. N. Cs *et al.*, "B4 : Experience with a Globally- Deployed Software Defined," 2013.

[71] N. McKeown *et al.*, "OpenFlow: enabling innovation in campus networks," *{ACM} {SIGCOMM} Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[72] ONF, "OpenFlow Switch Specification Version 1.0.0.0," vol. 0, pp. 1–42, 2009.

[73] J. S. Choi, H. P. Kim, M. S. Kim, M. S. Shin, J. M. Park, and E. Par, "An experimental implementation of MPLS-TP controlled Ethernet transport network for mobile backhaul," pp. 2–4.

[74] M. Azizi, R. Benaini, and M. Ben Mamoun, "Delay Measurement in Openflow-Enabled MPLS-TP Network," *Mod. Appl. Sci.*, vol. 9, no. 3, p. p90, Jan. 2015.

[75] Yang, K. Dong, L. Dong, and B. Li, *Research of the ARP spoofing principle and a defensive algorithm*, vol. 7, no. 5. WSEAS, 2008.

[76] Nippon Telegraph and Telephone Corporation, *Ryu Network Operating System*. 2012.

[77] P. A and J. Skalný, "OpenFlow 1.3 Software Switch by CPqD," 2014. [Online]. Available: http://cpqd.github.io/ofsoftswitch13/. [Accessed: 22-Dec-2018].

[78] B. Lantz, B. Heller, and N. Mckeown, "A Network in a Laptop : Rapid Prototyping for Software-Defined Networks," pp. 1–6, 2010.

[79] L. B. N. L. ESnet, "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool." [Online]. Available: https://iperf.fr/. [Accessed: 22-Dec-2018].

[80] Zheng Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, 1996.

[81] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," vol. 271, pp. 269–271, 1959.

[82] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theor. Comput. Sci.*, vol. 344, no. 2–3, pp. 243–278, 2005.

[83] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *J. Artif. Intell. Res.*, vol. 9, pp. 317–365, 1998.

[84] M. Azizi, R. Benaini, and M. B. Mamoun, "The programmable cloud network: Delay measurement application," in *Proceedings - 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2014*, 2015.

[85] "Traffic Monitor," 2018. [Online]. Available: http://osrg.github.io/ryu-book/en/html/traffic_monitor.html. [Accessed: 30-Sep-2019].

[86] Mininet, "Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet," *Mininet.Org*, 2014. [Online]. Available: http://mininet.org/. [Accessed: 05-Dec-2017].

[87] M. Azizi, R. Benaini, and M. Ben Mamoun, "MPLS-TP OAM Toolset: Interworking and Interoperability Issues," in *AFIN 2012, The Fourth International Conference on Advances in Future Internet*, 2012, pp. 60–64.

[88] "All-in-one SDN App Development Starter VM | SDN Hub." [Online]. Available: http://sdnhub.org/tutorials/sdn-tutorial-vm/. [Accessed: 17-Jan-2020].

Annexe A : mininet

Mininet est un logiciel open source qui est utilisé pour simuler les composants d'un réseau SDN: contrôleurs, commutateurs et hôtes. Par défaut, Mininet fournit des commutateurs OVS et des contrôleurs OVS interne. Cependant, il est possible d'installer d'autres contrôleurs et commutateurs SDN au lieu des contrôleurs et commutateurs par défaut. La principale caractéristique qui distingue les nœuds SDN des autres nœuds de commutation de données est qu'ils permettent de programmer les équipements réseau tels que les commutateurs et les routeurs et que le réseau se comporte en fonction des besoins des utilisateurs.

Mininet prend en charge le protocole Openflow, qui fournit une interface entre le plan de commande et le plan de transmission des données. Les protocoles Openflow sont utilisés pour contrôler le flux de paquets conformément à l'API écrite sur le contrôleur. Mininet prend également en charge diverses topologies et assure la disponibilité de topologies personnalisées. Mininet offre aussi la possibilité d'utilise un CLI (interface de ligne de commande) très confortable.

Installation de Mininet

Pour installer Mininet, de préférence sur une machine virtuelle Ubuntu, ouvrez le terminal et lancez la commande suivante :

```
# apt-get install mininet , pour installer les paquets Mininet sur votre système.
```

Ensuite, vérifiez l'installation en lançant la commande suivante :

```
# mn
```

Après une installation réussie, vous verrez à l'écran ce qui est montré dans la Figure suivante :

Remarquez qu'un réseau est créé avec une topologie par défaut et des commutateurs OVS par défaut. Ce réseau est prêt à l'emploi et dispose déjà de tous les paramètres comme les adresses IP et les liens préconfigurés, basés sur les paramètres par défaut.

Exemple de commandes Mininet

- La commande pour afficher les nœuds présents dans le réseau est :

```
mininet> nodes
```

- La commande ci-dessus va lister tous les nœuds présents dans le réseau créé. Comme le montre la Figure suivante, les nœuds s1, h1, h2 sont affichés.

- La commande permettant d'afficher et de lister les liens présents dans le réseau est la suivante

```
Mininet>net
```

```
mininet> nodes
available nodes are:
c0 h1 h2 s1
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet>
```

Comme le montre la figure précédente, l'interface eth0 de l'hôte h1 est connectée à eth1 du commutateur s1 et l'interface eth0 de l'hôte h2 est connectée à eth2 du commutateur s2.

- La commande pour afficher les adresses IP et les ID de processus des nœuds est :

```
Mininet>dump
```

Comme le montre la suivante, l'adresse IP 10.0.0.1 avec l'ID de processus 5118 est attribuée à h1 et l'adresse IP 10.0.0.2 avec l'ID de processus 5120 est attribuée à h2.

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=5118>
<Host h2: h2-eth0:10.0.0.2 pid=5120>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=5125>
<Controller c0: 127.0.0.1:6633 pid=5111>
mininet>
```

- La commande de ping d'un hôte spécifique vers un hôte cible est :

```
Mininet> h1 ping h2
```

```
mininet> h1 ping -c 3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=5.25 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.778 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.085 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.085/2.040/5.259/2.293 ms
mininet>
```

En inspectant les paquets, nous pouvons constater que le premier ping a pris beaucoup plus de temps (5,25 ms) que les autres. Ceci est dû au fait que les tables ARP, les tables MAC, etc. sont initialisées pendant le premier ping.

- La commande pour afficher les informations d'adresse des nœuds est :

```
Mininet> h1 ifconfig -a
```

Cette commande affiche l'adresse IP, l'adresse de diffusion et l'adresse MAC de l'hôte h1, comme sur la Figure suivante.

```
mininet> h1 ifconfig -a
h1-eth0   Link encap:Ethernet  HWaddr 7e:d9:f9:fd:29:ff
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:378 (378.0 B)  TX bytes:378 (378.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

mininet>
```

- La commande permettant de tester la connectivité entre les hôtes est

```
Mininet>pingall
```

Cette commande fera en sorte que chaque hôte du réseau effectue un ping sur tous les autres hôtes du réseau. Dans le réseau que nous avons, h1 fera un ping à h2, et h2 fera un ping à h1. Comme le montre la Figure suivante les pings réussis indiquent que tous les liens du réseau sont actifs.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

Lorsque nous étudions l'interface eth1 du commutateur s1 en utilisant Wireshark, nous constatons que les deux requêtes ping sont réussies (Figure 8).

- La commande pour couper un lien est :

```
Mininet> link s1 h1 down
```

La commande ci-dessus va couper la liaison entre le commutateur s1 et l'hôte h1 (Figure 9). De plus, en effectuant un ping sur les hôtes avec la commande pingall, nous pouvons voir que les deux pings échouent à cause de l'interruption de la liaison.

- La commande pour construire une topologie personnalisée est :

```
#sudo mn -topo simple,3
```

Cette commande va créer une topologie comme le montre la Figure 12, et initialiser les liens et les adresses des hôtes et des commutateurs.

- La commande pour effectuer un test de régression est :

```
#sudo mn -- test pingpair
```

La commande ci-dessus est utilisée pour créer un réseau avec une topologie par défaut, exécuter la fonction pingall et arrêter le réseau. Cette commande est essentiellement utilisée pour tester le fonctionnement de Mininet.

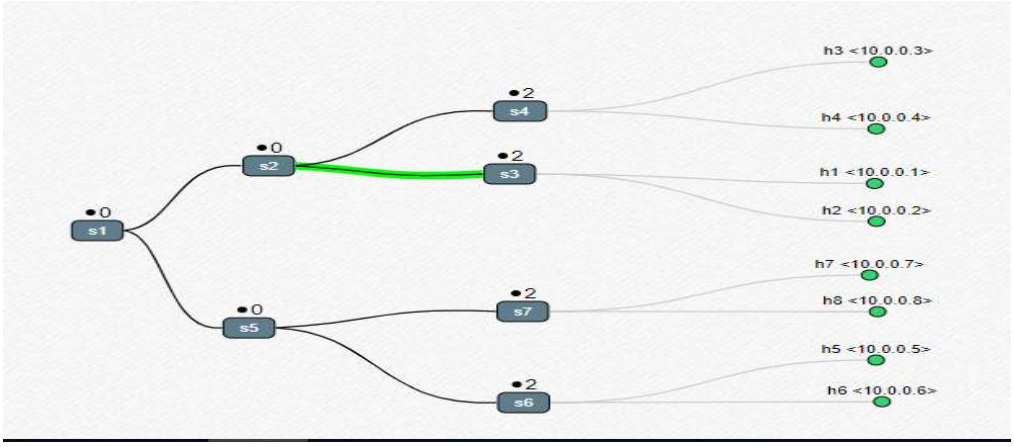
- La commande pour ouvrir la fenêtre xterm dans un environnement Linux est :

```
Mininet> h1 xterm
```

- La commande pour construire une topologie personnalisée est :

```
#sudo mn --topo=tree,3,2  --mac
```

Cette commande va créer une topologie comme le montre la figure suivante, et initialiser les liens et les adresses des hôtes et des commutateurs



- Dans l'exemple suivant, on lance une topologie mininet pour émuler un réseau MPLS avec un contrôleur externe Ryu :

```
mn  --custom  /root/mininet/custom/mpls.py  --topo=mpls  --mac  --switch
user,protocols=OpenFlow13 --controller remote,ip=192.168.1.101,port=6633
```

- au niveau de la console mininet, on peut altérer la valeur du délai en output sur le switch s0 port eth0 :

```
ethtool -K s0-eth1 gro off
tc qdisc del dev s0-eth1 root
tc qdisc add dev s0-eth1 root handle 10: netem delay 50ms
```

On a également la possibilité de générer du trafic UDP. Dans l'exemple suivant on génère 10 Mbps de flux UDP en utilisant l'utilitaire Iperf.

Au niveau client Iperf on précise le débit, le type de flux, l'adresse du serveur et la cadence d'envoi de trafic:

```
iperf -c 10.0.0.2 -u -b 10000k  -t 10
```

Au niveau serveur on exécute de la commande suivante:

```
iperf -s -i 1 -u
```

Annexe B : Contrôleur ryu

Ryu est un contrôleur SDN écrit en Python et développé par NTT. Il fournit des composants logiciels avec des API bien définies qui permettent aux développeurs de créer facilement de nouvelles applications de gestion et de contrôle de réseau. Ryu supporte divers protocoles pour la gestion des périphériques réseau, tels que OpenFlow, Netconf, OF-config, etc. A propos d'OpenFlow, Ryu supporte entièrement les extensions 1.0, 1.2, 1.3, 1.4, 1.5 et Nicira. Tout le code est disponible gratuitement sous la licence Apache 2.0.

Démarrage rapide

- Pour commencer, téléchargez et configurez le tutoriel SDN Hub VM dans Virtualbox ou VMware Player[88].
- Exécutez Mininet sur une fenêtre de terminal à l'aide de la commande suivante. Ceci démarre un environnement d'émulation réseau pour émuler 1 commutateur avec 3 hôtes.

```
 $ sudo mn --topo single,3 --mac --controller remote --switch ovsk
```

- La commande ci-dessus génère un commutateur qui supporte à la fois l'OpenFlow ver 1.0 et 1.3. Cependant, selon vos besoins, vous pouvez forcer un switch à supporter OpenFlow 1.3 en exécutant cette commande :

```
$ sudo ovs-vsctl set bridge s1 protocols=OpenFlow13
```

- Wireshark fait partie de la VM et peut analyser les messages OpenFlow 1.3. Pour démarrer Wirehark et visualiser les messages OpenFlow:

```
sudo wireshark &
```

- Ensuite, démarrez le contrôleur RYU. Supposons que le dossier principal où ryu est installé se trouve dans /home/ubuntu/ryu, La commande ci-dessous démarre le contrôleur en lançant l'application OpenFlow Protocol Handler et Simple Switch 1.3.
    - Comme le commutateur supporte OpenFlow 1.0 et 1.3, alors que l'application ne supporte que 1.3, le système va s'auto-négocier et choisir de procéder à OpenFlow 1.3.

```
$ cd /home/ubuntu/ryu && ./bin/ryu-manager --verbose
ryu/app/simple_switch_13.py
loading app ryu/app/simple_switch_13.py
loading app ryu.controller.ofp_handler
instantiating app ryu/app/simple_switch_13.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
```

```
BRICK SimpleSwitch13
    CONSUMES EventOFPPacketIn
    CONSUMES EventOFPSwitchFeatures
BRICK ofp_event
    PROVIDES EventOFPPacketIn TO {'SimpleSwitch13': set(['main'])}
    PROVIDES EventOFPSwitchFeatures TO {'SimpleSwitch13': set(['config'])}
    CONSUMES EventOFPErrorMsg
    CONSUMES EventOFPSwitchFeatures
    CONSUMES EventOFPPortDescStatsReply
    CONSUMES EventOFPEchoRequest
    CONSUMES EventOFPHello
```

- Assurez-vous de démarrer le bon ryu-manager si plusieurs versions du contrôleur sont installées dans le même système.
- Ensuite, vérifiez si les hôtes dans la topologie mininet peuvent se rejoindre

```
mininet> h1 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_req=1 ttl=64 time=2.76 ms
64 bytes from 10.0.0.3: icmp_req=2 ttl=64 time=0.052 ms
64 bytes from 10.0.0.3: icmp_req=3 ttl=64 time=0.051 ms
```

Structure du Code Ryu

Le code du contrôleur principal est organisé sous le dossier /ryu/ (Dans notre VM - /home/ubuntu/ryu/ryu/). Ici, nous discutons des fonctionnalités des composants clés. Il est important de se familiariser avec eux.

- app/ - Contient un ensemble d'applications qui s'exécutent sur l'API nord du contrôleur.
- base/ - Contient la classe de base pour les applications RYU. La classe RyuApp dans le fichier app_manager.py est héritée lors de la création d'une nouvelle application.
- controller/ - Contient l'ensemble des fichiers nécessaires pour gérer les fonctions OpenFlow (par exemple, les paquets des commutateurs, la génération des flux, la gestion des événements réseau, la collecte des statistiques etc).
- lib/ - Contient un ensemble de bibliothèques de paquets pour analyser les différents en-têtes de protocole et une bibliothèque pour OFConfig. De plus, il inclut également des analyseurs pour Netflow et sFlow.
- ofproto/ - Contient les informations spécifiques au protocole OpenFlow et les analyseurs associés pour supporter les différentes versions du protocole OF (1.0, 1.2, 1.3, 1.4)

- topologie/ : Contient le code qui effectue la découverte de la topologie liée aux commutateurs OpenFlow et gère les informations associées (par exemple, les ports, les liens, etc.). on utilise le protocole LLDP en interne.

Principes essentiels du code du contrôleur RYU

La plupart des plateformes de contrôleurs exposent certaines fonctionnalités natives pour permettre ces fonctionnalités clés :

- Possibilité d'écouter les événements asynchrones (par exemple, PACKET_IN, FLOW_REMOVED) et d'observer les événements en utilisant le décorateur ryu.controller.handler.set_ev_cls.
- Possibilité d'analyser les paquets entrants (par exemple, ARP, ICMP, TCP) et de fabriquer des paquets à envoyer sur le réseau
- Possibilité de créer et d'envoyer un message OpenFlow/SDN (par exemple, PACKET_OUT, FLOW_MOD, STATS_REQUEST) au plan de données programmable.

Avec RYU, nous pouvons réaliser tout cela en invoquant un ensemble d'applications pour gérer les événements réseau, analyser toute demande de commutation et réagir aux changements du réseau en installant de nouveaux flux, si nécessaire. Par exemple, la création d'une nouvelle application implique la création d'une sous-classe de RyuApp et la construction de la logique requise pour écouter les événements réseau.

```
from ryu.base import app_manager


class L2Forwarding(app_manager.RyuApp):
    def __init__(self, *args, **kwargs):
        super(L2Forwarding, self).__init__(*args, **kwargs)
```

Bien que le code ci-dessus représente une application RYU valide, il n'a pas la logique pour gérer les événements réseau provenant des commutateurs OpenFlow. Ensuite, pour permettre à une application de recevoir les paquets envoyés par le commutateur au contrôleur, la classe doit implémenter une méthode qui est décorée par EventOFPPacketIn.

```
    @set_ev_cls(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER)
    def packet_in_handler(self, ev):
```

Le premier argument du décorateur appelle cette fonction à chaque fois qu'un message packet_in est reçu. Le second argument indique l'état du switch. Toute information concernant le switch et la version du protocole supporté par le switch peut être déchiffrée en utilisant ce qui suit :

```
    msg = ev.msg
    datapath = msg.datapath    # Switch Datapath ID
    ofproto = datapath.ofproto
```

Une fois que le paquet est reçu, nous pouvons le décoder en important la bibliothèque de paquets sous /ryu/lib :

```
    from ryu.lib.packet import packet
    from ryu.lib.packet import ethernet
```

Nous pouvons inspecter les en-têtes de paquets pour plusieurs types de paquets : ARP, Ethernet, ICMP, IPv4, IPv6, MPLS, OSPF, LLDP, TCP, UDP.

```
    pkt = packet.Packet(msg.data)
    eth = pkt.get_protocol(ethernet.ethernet)
```

Nous utilisons les deux commandes suivantes pour extraire les détails de l'en-tête de l'Ether :

```
    dst = eth.dst
    src = eth.src
```

De même, la classe OFPPacketOut peut être utilisée pour construire un message packet_out avec les informations requises (par exemple, l'ID du chemin de données, les actions associées, etc

```
    out                                                                 =
ofp_parser.OFPPacketOut(datapath=dp,in_port=msg.in_port,actions=actions)#
Generate the message
    dp.send_msg(out)
```

Outre le PACKET_OUT, nous pouvons également effectuer une insertion FLOW_MOD dans un switch. Pour cela, nous construisons le Match, l'Action, les Instructions et générons le Flux requis. Voici un exemple de création d'un en-tête de match où les matchs in_port et eth_dst sont extraits du PACKET_IN :

```
    msg = ev.msg
    in_port = msg.match['in_port']
    # Get the destination ethernet address
    pkt = packet.Packet(msg.data)
    eth = pkt.get_protocol(ethernet.ethernet)
    dst = eth.dst
    match = parser.OFPMatch(in_port=in_port, eth_dst=dst)
```

Voici un exemple de création d'une liste d'actions pour le flux.

```
  actions = [ofp_parser.OFPActionOutput(ofp.OFPP_FLOOD)]
```

OpenFlow 1.3 associe à chaque entrée de flux un ensemble d'instructions telles que des actions de traitement pour modifier/transférer des paquets, des instructions de traitement de pipeline de support dans le cas de multi-tables, des instructions de comptage pour limiter le trafic, etc. L'ensemble des actions précédentes définies dans OpenFlow 1.0 sont un type d'instructions défini dans OpenFlow 1.3.

Une fois que la règle de correspondance et la liste d'actions sont formées, les instructions sont créées comme suit :

```
    inst                                                                    =
[parser.OFPInstructionActions(ofproto.OFPIT_APPLY_ACTIONS, actions)]
```

Compte tenu du code ci-dessus, un flux peut être généré et ajouté à un commutateur particulier.

```
    mod = parser.OFPFlowMod(datapath=datapath, priority=0, match=match,
instructions=inst)
    datapath.send_msg(mod)
```

Exemple de code de l'algorithme Dijkstra

Ci-dessous on presente le bout de code de l'algorithme Dijkstra utilisant l'API nord du contrôleur Ryu:

```
def distance_minimale(distance, C):
  min = float('Inf')
  node = 0
  for v in C:
    if distance[v] < min:
      min = distance[v]
      node = v
  return node

def get_chemin_Dijkstra (src,dst,first_port,final_port):

  distance = {}
  precedent = {}

  for dpid in commutateurs:
    distance[dpid] = float('Inf')
    precedent[dpid] = None
    distance[src]=0
    C=set(commutateurs)
    print "C=", C
    while len(C)>0:
            u    =    distance_minimale(distance,    C)        //Appel    de
distance_minimale
            C.remove(u)
    for p in commutateurs:
      if adjacency[u][p]!=None:
        w = 1
```

```
        if distance[u] + w < distance[p]:
          distance[p] = distance[u] + w
          precedent[p] = u
  r=[]
  p=dst
  r.append(p)
  q=precedent[p]
  while q is not None:
    if q == src:
      r.append(q)
      break
      p=q
    r.append(p)
    q=precedent[p]

  r.reverse()
  if src==dst:
    path=[src]
  else:
    path=r

   # Ajout des ports
  r = []
  in_port = first_port
  for s1,s2 in zip(path[:-1],path[1:]):
    out_port = adjacency[s1][s2]
    r.append((s1,in_port,out_port))
    in_port = adjacency[s2][s1]
  r.append((dst,in_port,final_port))
  return r
```

L'exécution du code se fait de la manière suivante:

```
cd /home/ubuntu/ryu && ./bin/ryu-manager --verbose --observe-links
./app/dijkstra_ryu.py ./topology/switches.py ./topology/ofctl_rest.py
```

Résumé

Les réseaux de transports connaissent depuis plusieurs années un changement capital en passant de technologies traditionnelles à commutation par circuit (TDM) à des réseaux dit de nouvelle génération (NGN) comme Multiprotocol Label Switching Transport Profile (MPLS-TP). Toutefois différents standards d'outils d'Opérations, Administrations et Maintenances (OAM) coexistent ensemble ce qui pose de réel problèmes aux opérateurs de transports réseaux.

Cette thèse présente d'abord les différents outils OAM utilisées dans les réseaux MPLS-TP. Ensuite, elle expose la problématique d'interopérabilité des OAM causée par le fait qu'il y a deux familles de standards différents d'OAM. Après avoir fait l'état de l'art des solutions traitant cette problématique, nous avons proposé deux solutions, intitulées « modèle overlay » et « modèle de cloisonnement », pour résoudre cette problématique Par la suite, nous avons appliqué le paradigme Software Defined Netwrking (SDN) aux réseaux MPLS-TP pour proposer d'exécuter les OAM comme un service réseau. En dernière partie, nous proposons une solution pour utiliser des outils OAM afin d'optimiser les ressources dans des réseaux de transport MPLS-TP basée sur le paradigme SDN. Pour se faire, notre approche se base sur l'utilisation de l'algorithme de colonie de fourmis modifié qui est comparée à d'autres approches existantes. A la fin de ce manuscrit, l'environnement, la méthodologie et les résultats des simulations sont analysées.

Mots-clefs: MPLS-TP, OAM, SDN, Openflow, Algorithme de Colonie de Fourmis.

Abstract

During last years, transport networks have been undergoing a major change by moving from traditional circuit-switched (TDM) technologies to so-called New Generation Networks (NGN) like Multiprotocol Label Switching Transport Profile (MPLS-TP). However, different standards of Operations, Administration and Maintenance (OAM) tools coexist together, which poses real problems for network transport operators.

This thesis first presents the different OAM standard and highlights their interoperability's problem. Then, we have proposed two solutions resolve this problems the first is called "overlay model" and the second is called "partitioning model". We then applied the Software Defined Networking (SDN) paradigm to MPLS-TP networks in order to launch OAM directly as a network service. Finally, we propose a solution to use OAM tools to optimize resources in MPLS-TP transport networks based on the SDN paradigm using a modified ant colony algorithm which is compared to other existing approaches. At the end of this manuscript, the simulation results are extensively detailed and analyzed.

Key Words: MPLS-TP, OAM, SDN, Openflow, Ant-Colony Algorithm