

Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : **Télécommunications**

Spécialité : **Réseaux et télécommunications**

Thème

Mise en place un réseau SDN (software-Defined Networking)

Préparé par :

- Berbache Yuba
- Benouaret Nadir

Dirigé par :

- M. O Bessaad
M. I.Herroug

Examiné par :

- Mme Gherbi
Mme Ghennam (P)

REMERCIEMENTS

Nous remercions d'abord dieu, qui nous a donné ses facultés morales et intellectuelles pour penser, réfléchir, observer et concrétiser ce travail avec une large patience. C'est d'ailleurs, grâce à dieu qui nous a entourées d'aimable personnes compréhensible et coopératives, qui nous avons mené à bien notre tâche.

Je voudrais dans un premier temps remercier, mon directeur de mémoire Mr. BESSAAD Omar, mon professeur à l'université Abderrahmane mira Bejaia, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion

Aussi à tous enseignant et employés du département d'Automatique, Télécommunication et d'Electronique.

Je tiens à remercier Mr HAROUG Aimad et Mr HIDOUS Abdelhakim nos encadreurs de l'entreprise IFR1 ont contribué au succès de mon stage et qui m'ont aidée lors de la rédaction de ce mémoire.

Nos vifs remerciements s'adressent également à Monsieur le président du jury et les membres de jury pour l'honneur de jury ce travail.

Mes parents, pour les soutiens constants et leurs encouragements.

Yuba&Nadir

DÉDICACES

Je dédie ce précieux travail

À mes chers parents ; Ce travail est bien plus qu'un simple projet, je le dois à mes très chers parents qui m'ont fourni chaque jour un soutien indéfectible et une confiance inébranlable. Ils m'ont offert un modèle exemplaire de travail et de persévérance. J'espère qu'ils trouveront dans ce travail l'expression de ma profonde gratitude et de tout mon amour. Que Dieu vous protège.

À ma sœur Hanane ; Les mots ne suffisent pas à exprimer le respect profond et l'admiration que je te porte. Merci pour ta patience, tes conseils avisés, et ton soutien constant. Je t'aime énormément.

À mon frère Salah ; Tu as toujours été à mes côtés, et pour cela, je te suis infiniment reconnaissant.

À mes chers amis ; Samy, Anis, Yahia, et Abderahmane. Merci pour tous les moments agréables que nous avons partagés

À mon binôme Yuba ; Je dédie ce travail qui est le fruit de notre effort commun et sans qui il n'aurait jamais vu le jour. Merci infiniment pour ta collaboration et ton soutien.

Nadir

DÉDICACES

Je dédie ce précieux travail

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

À ma sœur Thiziri ; Ta patience, tes conseils avisés et ton soutien constant m'ont été d'une aide précieuse. Ta présence apaisante et ton écoute ont souvent été mon refuge. Je te suis profondément reconnaissant et je t'aime énormément.

À mon frère Arezki ; Malgré la distance qui nous sépare, ta présence a toujours été ressentie. Ton soutien depuis la France m'a été d'un grand réconfort. Merci pour ton encouragement constant et ta fierté, qui m'ont inspiré à donner le meilleur de moi-même.

À mes chers amis ; Yahia, Anis, Samy, Abderahmane, Chaque moment passé avec vous a été une source de bonheur et de motivation.

À mon binôme nadir ; Ce travail n'aurait jamais été possible sans ta collaboration et ton soutien. Ton dévouement et ta coopération ont été essentiels à notre réussite commune. Merci infiniment pour ton engagement et ton amitié

Yuba

Table de matière

<i>Table de matière</i>	5
<i>Liste des figures</i>	14
<i>Liste des tableaux</i>	14
<i>Liste d'abréviations</i>	13
<i>Introduction Générale</i>	1
<i>Chapitre 1 : Etude de l'existant</i>	2
1.1. <i>Introduction</i>	2
1.2. <i>Présentation de l'organisme d'accueil</i>	2
1.2.1. <i>Présentation de l'entreprise Bejaia Logistique</i>	2
1.3. <i>Historique de Bejaia Logistique</i>	1
1.4. <i>Cadre juridique de Bejaia Logistique</i>	2
1.5. <i>Les activités de Bejaia Logistique</i>	4
1.6. <i>Infrastructure informatique de Bejaia logistique</i>	3
1.7. <i>Structure de Bejaia Logistique</i>	4
1.7.1. <i>L'organigramme générale</i>	4
1.8. <i>La structure organisationnelle</i>	6
1.8.1. <i>La direction générale</i>	6
1.8.2. <i>Département secrétariat de direction</i>	7
1.8.3. <i>Département informatique</i>	5
1.8.4. <i>Département RH</i>	5
1.8.5. <i>Département facturation et comptabilité</i>	5
1.8.6. <i>Département commercial</i>	5
1.8.7. <i>Département d'exploitation</i>	6
1.9. <i>Diagramme réseau informatique de BL</i>	7
1.10. <i>Explication détaillée du diagramme réseau</i>	8
1.11. <i>Fonctionnement du réseau</i>	8
1.12. <i>Architectures du réseau informatique de GROUPE IFRI</i>	10
1.13. <i>Description de l'architecteur</i>	11
1.14. <i>Découverte du Serveur Central et de l'Infrastructure Réseau de l'Entreprise</i>	12
1.15. <i>Critique sur l'architecture Actuelle du Groupe IFRI</i>	13
1.16. <i>Étude de l'Architectures Réseau de Bejaia Logistique</i>	13
1.17. <i>Architectures Réseau de BL</i>	14
1.18. <i>Caractéristiques des équipements utilisés dans le réseau de BL</i>	14
1.19. <i>Problématique</i>	17
1.20. <i>Les Solutions proposée</i>	18
1.20.1. <i>Présentation de la première solution</i>	18

1.20.2.	<i>Inconvénients de l'architecture réseau proposée 1</i>	17
1.20.3.	<i>Présentation de deuxième solution</i>	18
1.20.4.	<i>Inconvénients de l'architecture réseau proposée 2</i>	19
1.20.5.	<i>Présentation de la troisième solution</i>	21
1.21.	<i>Solution retenue</i>	22
1.22.	<i>Objectif et résultat visés</i>	22
1.23.	<i>Conclusion</i>	22
<i>Chapitre 2 : Réseau défini par l'ogiciel SDN</i>		24
2.1.	<i>Introduction</i>	24
2.2.	<i>Principe de réseau SDN</i>	24
2.2.1.	<i>Réseau traditionnelle vers réseau SDN</i>	24
2.2.2.	<i>Problématiques des réseaux traditionnels</i>	25
2.2.3.	<i>La comparaison entre réseau traditionnel et SDN</i>	26
2.2.4.	<i>Historique</i>	27
2.2.5.	<i>Définition du SDN</i>	29
2.2.6.	<i>C'est quoi SDN</i>	29
2.2.7.	<i>Architecture de SDN</i>	30
2.2.7.1	<i>SDN se compose de 3 couche</i>	30
2.2.8.	<i>Domaines d'application le réseau SDN</i>	33
2.2.9.	<i>Composants SDN</i>	34
2.2.9.1.	<i>Contrôleur SDN</i>	34
2.2.9.2.	<i>Commutateur SDN</i>	39
2.2.9.3.	<i>Commutateur SDN</i>	40
2.2.10.	<i>Établissement d'une connexion commutateur-contrôleur</i>	42
2.2.11.	<i>Processus de transmission d'un paquet avec OpenFlow</i>	44
2.2.12.	<i>Les Tables OpenFlow</i>	45
2.2.12.1.	<i>Les Tables de Flux</i>	45
2.2.12.2.	<i>Les principaux éléments d'une entrée dans une table de Flux</i>	45
2.2.13.	<i>Processus de traitement d'un paquet dans un Switch OpenFlow</i>	46
2.2.14.	<i>Les Tables de groupe</i>	49
2.2.15.	<i>Message OpenFlow</i>	49
2.2.15.1.	<i>Les Messages depuis le Contrôleur vers Switch</i>	49
2.2.16.	<i>Opportunités et défis des SDN</i>	51
2.2.17.	<i>Les défis des réseaux SDN</i>	52
2.2.17.1.	<i>La performance</i>	52
2.2.17.2.	<i>La scalabilité</i>	52
2.2.17.3.	<i>La fiabilité</i>	52
2.2.17.4.	<i>La sécurité</i>	52
2.2.18.	<i>Impacte des réseaux SDN</i>	53
2.3.	<i>Conclusion</i>	53
<i>Chapitre 3 : La mise en place et Résultats</i>		54
3.1.	<i>Introduction</i>	54
3.2.	<i>Architecture proposée</i>	54

3.3.	<i>Outils et Technologies</i>	55
3.4.	<i>Taches</i>	56
3.4.1.	<i>Présentation de l'environnement de travail</i>	56
3.4.1.1.	<i>Installation de VMWare Workstation pro</i>	56
3.4.1.2.	<i>Configurer la machine virtuelle Mininet</i>	58
3.4.1.3.	<i>Installez opendaylight sur ububtu 20.04</i>	62
3.4.1.4.	<i>Création de la topologie SDN</i>	69
3.5.	<i>Conclusion</i>	84
	<i>Conclusion Générale</i>	85
	<i>Bibliographie</i>	86

Table de figures

Figure 1- Logo de Bejaia Logistique	1
Figure 2- fiche signalétique du Groupe IFRI	2
Figure 3– organigramme générale de l’entreprise BL.....	4
Figure 4– Architecture de réseaux de BL.....	7
Figure 5- Architecture actuel du réseau informatique de GROUPE IFRI.....	10
Figure 6- D-LINK DIR-685 Wireless AC Routeur.....	11
Figure 7- Cisco Catalyst 9120axi.....	13
Figure 8- PC standard.....	12
Figure 9- Centre Data IFRI.....	12
Figure 10- Architecture actuel du réseau informatique BL.....	14
Figure 11- Architecture proposée 1.....	17
Figure 12- Architecteur proposée 2.....	18
Figure 13- architecteur proposée 3.....	19
Figure 14- Impacts et conséquences de l’architecture réseau traditionnel.....	24
Figure 15- Comparaison entre les contrôles traditionnels et centralisés d’un SDN.....	25
Figure 16- le développement d’un réseau SDN.....	26
Figure 17- Architecture d’un routeur.....	27
Figure 18- Architecteur SDN.[23].....	28
Figure 19– Processus de communication entre le Switch et le contrôleur.....	30
Figure 20- Communication inter-contrôleur dans une architecteur distribuée.....	31
Figure 21- Fonctionnement Dun contrôleur SDN.....	33
Figure 22– Logo du contrôleur NOX.....	33
Figure 23– Logo du contrôleur POX.....	34

Figure 24– Logo du contrôleur Beacon.....	34
Figure 25– Logo du contrôleur Floodlight.....	34
Figure 26– Logo du contrôleur OpenDaylight.....	35
Figure 27- Architecteur orchestrateur SDN.....	36
Figure 28 : open vswitch.....	38
Figure 29- logo protocole openflow.....	38
Figure 30- Architecture OpenFlow.....	40
Figure 31- Connexion au contrôleur OpenFlow.....	41
Figure 32- Échec de la connexion au contrôleur OpenFlow.....	41
Figure 33- Mode d'urgence.....	42
Figure 34- Processus de transmission des paquets au sein des commutateurs.....	43
Figure 35- Contenu d'entrées de flux.....	44
Figure 36- Processus de traitement d'un paquet.....	44
Figure 37- Traitement d'un paquet.....	46
Figure 38- architecteur test de connectivité.....	48
Figure 39: la topologie SDN mise en place.....	52
Figure 40 :présente L’interface graphique de VMware Workstation pro.....	55
Figure 41 : Téléchargement Mininet.....	56
Figure 42: Créer la machine virtuelle Mininet.....	57
Figure 43: ouvrir Mininet.....	57
Figure 44: configuration Mininet.....	58
Figure 45 :Démarrer Mininet.....	58
Figure 46 :Tester Mininet.....	59
Figure 47:adresse Mininet.....	59
Figure 48:télécharger l’image iso d’ubuntu server.....	60
Figure 49:Creation une machine vitrual odl.....	60

Figure 50 configuration ODL	61
Figure 51 : affichage adresse ip Mininet	61
Figure 52 : session ODL	62
Figure 53 : adresse IP ODL	63
Figure 54 Démarrer.opendaylight	65
Figure 55 : Accès à l'interface d'utilisateur dlux.opendaylight	66
Figure 56 : l'interface dlux.opendaylight	67
Figure 57 : L'adresse ip de mininet VM	68
Figure 58 :L'adresse ip de ODL VM	68
Figure 59 : script python	70
Figure 60 : Exécution d script python	70
Figure 61 : Tester la connectivité	71
Figure 62 : topologie.opendaylight	72
Figure 63 : ping hote 1 vers hote 2	73
Figure 64 : ping hote 1 vers hote 2 seulement pour 3 premier paquets	73
Figure 65: Liste des nodes	74
Figure 66 : Affichage des nodes sur mininet vm	74
Figure 67 : Affichage l'état du réseau simulé	75
Figure 68 : Affichage la table ds flux de switch 1	75
Figure 69 : Table de flux SW.OF.1	76
Figure 70 :Table de flux SW.OF.2	76
Figure 71 : Table de flux SW.OF3	77
Figure 72: Table de flux SW.OF4	77
Figure 73 :: Table de flux SW.OF5	78
Figure 74 : Table de flux SW.OF6	78

Figure 76: Types de requêtes REST API.....	80
Figure 77:Vue d'ensemble de POSTMAN.....	80
Figure 78: l'authentification.....	81
Figure 79: En tête d'une requête PUT.....	82
Figure 80 : Exemple de script XML.....	83
Figure 81 : Résultat de la requette envoyer.....	83

Liste des tableaux

Tableau 1- caractéristiques des équipements utilisés dans le réseau de BL	15
Tableau 2- tableau comparatif entre le SDN est les réseau traditionnels	25
Tableau 1 : Informations Hôtes.....	92

Liste d'abréviations

API:	Application Programming Interface
CLI:	Command Line Interface
CPU:	Central Processing Unit
DARPA:	Defense Advanced Research Project Agency
DNS:	Domain Name System
DPI:	Deep Packet Inspection
GUI:	Graphical User Interface
HTTP:	HyperText Transfer Protocol
HTTPS:	HyperText Transfer Protocol Secure
ICMP:	Internet Control Message Protocol
IP:	Internet Protocol
LAN:	Local Area Network
MAN:	Metropolitan Area Network
NFV:	Network Function Virtualization
OSI:	Open Systems Interconnection
RESTCONF:	RESTful Configuration Protocol
REST :	REpresentationnal State Transfer
SDN:	Software-Defined Networking
OVS:	Open Virtual Switch
SSH :	Secure Shell
SMTP:	Simple Mail Transfer Protocol
TCP:	Transmission Control Protocol
TCP/IP:	Transmission Control Protocol/Internet Protocol
UDP:	User Datagram Protocol
VMM:	Virtual Machine Monitor
VM:	Virtual Machine
VLAN:	Virtual Local Area Network
VSAT:	Very Small Aperture Terminal
WAN:	Wide Area Network
NVR:	Network Video Recorde

Introduction Générale

Ce mémoire explore la mise en place d'un réseau SDN au sein de Bejaia Logistique, une entreprise confrontée à des défis croissants liés à la gestion de ses infrastructures réseau traditionnelles.

Le SDN, ou Software-Defined Networking, propose une gestion dynamique et centralisée des réseaux via des logiciels, séparant le plan de contrôle du plan de données. Et les réseaux traditionnels, souvent rigides et complexes, peinent à s'adapter aux besoins changeants des entreprises modernes.

Bejaia Logistique, comme beaucoup d'autres, fait face à des problèmes de scalabilité, de performance et de sécurité. Le SDN apparaît comme une solution prometteuse, offrant une gestion plus agile et centralisée des ressources réseau.

Notre intérêt pour les réseaux SDN découle de notre passion pour les technologies de l'information et de la communication. Après avoir constaté les limites des réseaux traditionnels au cours de divers stages et projets, je suis convaincu que le SDN représente l'avenir des infrastructures réseau. Ce mémoire est une opportunité de contribuer à l'innovation technologique chez Bejaia Logistique. Avec l'augmentation exponentielle de l'utilisation des technologies de l'information, les réseaux traditionnels montrent leurs limites, étant complexes à configurer et difficiles à faire évoluer. La gestion des données et des services devient de plus en plus ardue avec les outils traditionnels. Face à ces défis, les chercheurs ont développé le SDN, une approche qui centralise et simplifie la gestion des réseaux. Le SDN permet aux administrateurs de contrôler et d'automatiser le réseau via une interface logicielle, offrant ainsi une flexibilité et une agilité accrues pour une gestion plus efficace des ressources réseau.

L'objectif de ce travail est de présenter et d'identifier ce paradigme incontournable pour les réseaux du futur : le réseau programmable ou SDN. Ce concept ouvre la voie à des réseaux plus intelligents et flexibles, prêts à répondre aux besoins des applications commerciales futures.

Ce mémoire est structuré en quatre chapitres :

Présentation de l'organisme d'accueil : Introduction à Bejaia Logistique et ses défis actuels.

Généralités sur les réseaux informatiques : Vue d'ensemble des réseaux traditionnels et des problématiques qu'ils posent.

Principe de réseau SDN : Exploration détaillée du SDN, de ses composants et de ses avantages.

Implémentation et Simulation d'un réseau SDN : Mise en œuvre pratique d'une solution SDN.

Chapitre 1

Etude de l'existant

1.1. Introduction

L'étude de l'existant constitue une étape cruciale dans la mise en place d'un projet de réseau SDN (Software-Defined Networking). Dans ce chapitre, nous poserons la problématique de l'entreprise après nous allons nous pencher sur la SARL Bejaia Logistique, en commençant par une présentation et l'historique de l'entreprise, depuis sa création jusqu'à ses activités actuelles. Nous étudierons également la situation informatique de l'entreprise, en examinant les outils et les technologies utilisés pour optimiser ses opérations logistiques. Enfin, nous analyserons les besoins spécifiques de l'entreprise et proposerons des solutions adaptées et efficaces pour la transition vers un réseau SDN.

1.2. Présentation de l'organisme d'accueil

1.2.1. Présentation de l'entreprise Bejaia Logistique

La SARL Bejaia Logistique a été fondée en 2008 et est reconnue comme l'une des principales entreprises algériennes spécialisées dans le transport routier. Sa réputation de qualité et sa notoriété nationale sont le fruit de son important parc de transport, ainsi que de ses offres de location d'engins et de matériel pour les travaux publics, la manutention, la location de véhicules avec ou sans chauffeur, et le transport de produits pétroliers. Elle est enregistrée auprès du registre de commerce sous le numéro 07B0185663. Bejaia Logistique est implantée dans la zone industrielle AHRIK, IGHZER AMOKRAN, située dans la commune d'Ouzellaguen, dans la wilaya de Bejaia, au nord-est de l'Algérie. Elle possède un capital de 95 400 000 DA et a réalisé un chiffre d'affaires de 1 940 619 000 DA en 2017. L'entreprise se concentre sur la fourniture de solutions logistiques dans divers domaines, ce qui explique la multiplication de ses clients internes et externes. [1]

1.3. Historique de Bejaia Logistique

Au départ, avant d'obtenir son statut juridique de SARL, BL (Bejaia Logistique) était un service de parc et de transport dans l'entreprise de production d'eau minérale et de boissons diverses appelée SARL Ibrahim et Fils "Ifri". La création de ce service remonte à 2002, avec pour mission de transporter les marchandises produites par l'entreprise dans tout le pays. Au fil des années, la production d'Ifri a augmenté, mais son système de distribution a été confronté à de nombreux problèmes. Les couts de maintenance de ses moyens de transport étaient également très élevés, en particulier pendant la période hivernale. Pour alléger cette charge et éviter les couts associés à l'utilisation de ses propres véhicules, Ifri a décidé de centraliser son service de parc et de transport.

Cette entreprise a été nommée Bejaia Logistique (BL) et a été créée en octobre 2008. Au début, BL n'était qu'une petite entreprise chargée uniquement du transport des marchandises de sa société mère. Cependant, au fil des ans, elle a connu un grand succès et a vu son activité évoluer au fur et à mesure de sa performance. Grâce à une excellente gestion, elle est devenue une entreprise indépendante qui agit et pense par elle-même. Elle est passée d'une petite entreprise en 2008 à une moyenne entreprise, puis à une grande entreprise en seulement 10 ans.[1]



Figure 1- Logo de Bejaia Logistique . [1]

1.4. Cadre juridique de Bejaia Logistique

Numéro de registre de commerce est : 98B0182615 ;

Numéro d'article d'imposition : 06360646615 ;

Numéro d'identifiant fiscale : 099806018261598.



Figure 2- fiche signalétique du Groupe IFRI

1.5. Les activités de Bejaia Logistique

La Sarl BL offre un large éventail de services, notamment le transport public de marchandises, la location d'engins et de véhicules avec ou sans chauffeur, ainsi que la location de matériel pour les bâtiments et les travaux publics. Au quotidien, BL se fixe de nombreuses missions liées à ses activités, telles que : Garantir la satisfaction des clients. Assurer la sécurité des personnes et des biens en relation avec les activités de transport. Gérer la conduite, l'exploitation et la maintenance de ses réseaux d'activités, et étendre ses activités dans des zones inexplorées hors du pays.

1.6. Infrastructure informatique de Bejaia logistique

L'entreprise d'accueil dispose d'un ensemble complet d'infrastructures informatiques pour soutenir ses activités. Au total, nous avons 08 ordinateurs, répartis dans différents départements et services. Ces ordinateurs sont soigneusement sélectionnés pour répondre aux besoins spécifiques de chaque équipe.

En ce qui concerne les logiciels, l'entreprise utilise au total 18 applications et programmes informatiques pour faciliter ses opérations. Parmi ces logiciels, 3 ont été développés en interne, spécialement conçus pour répondre aux exigences particulières de l'entreprise. En ce qui concerne l'infrastructure matérielle, l'entreprise dispose de 09 imprimantes pour répondre aux besoins d'impression de ses employés. Ces imprimantes sont stratégiquement réparties dans les différents bureaux et départements pour garantir une accessibilité facile à tous. Pour assurer une connectivité réseau fiable, l'entreprise possède 13 équipements réseau, tels que des routeurs, des commutateurs et des ponts d'accès sans fil. Ces équipements assurent une communication fluide entre les ordinateurs et les autres périphériques connectés au réseau.

En ce qui concerne la sécurité, l'entreprise dispose de 7 caméras de surveillance qui sont installées à des endroits clés de l'entreprise pour assurer la sécurité des locaux et des biens.

1.7. Structure de Bejaia Logistique

1.7.1. L'organigramme générale

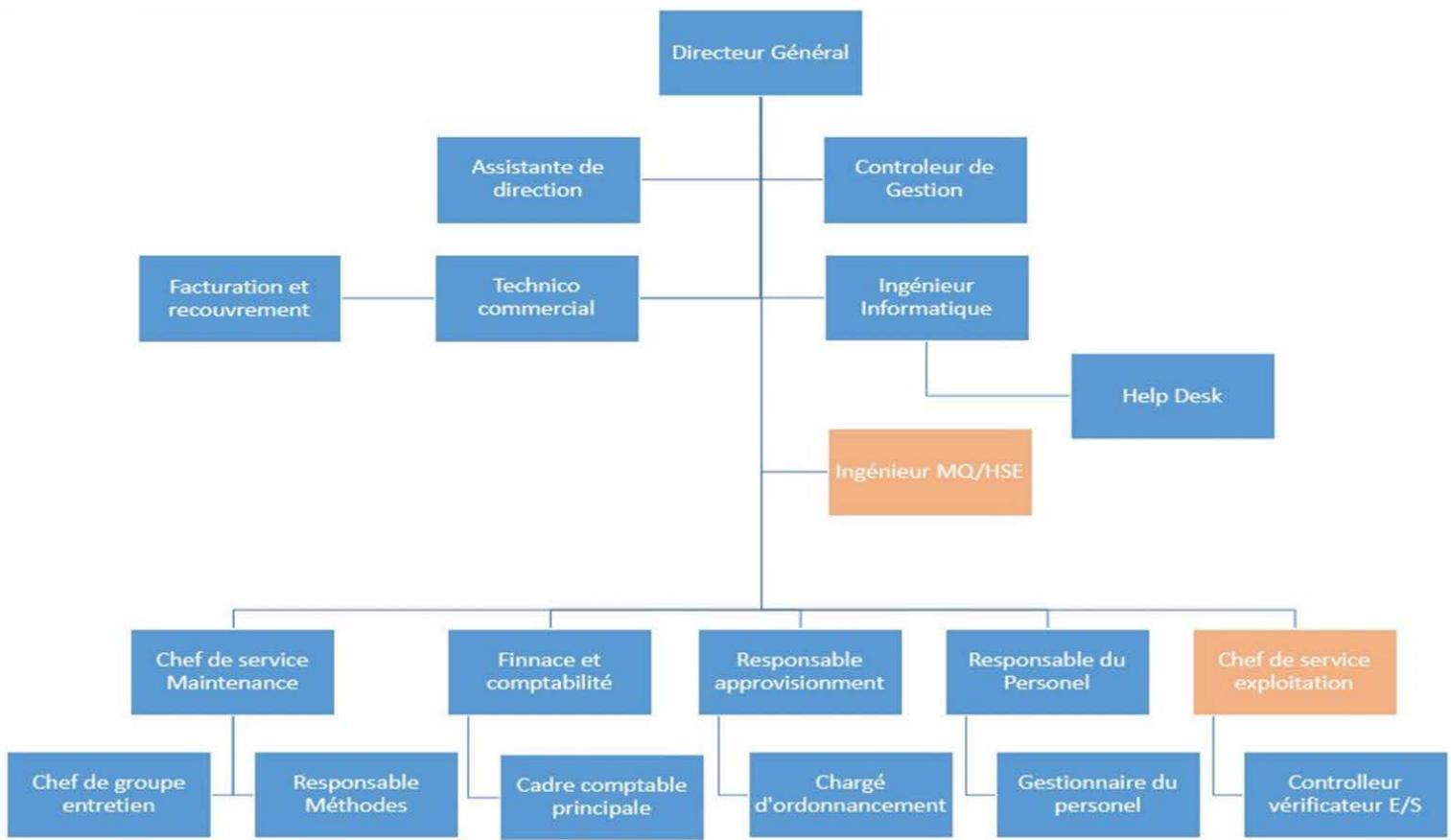


Figure 3- organigramme générale de l'entreprise BL

1.8. La structure organisationnelle

1.8.1. La direction générale

Elle veille sur le bon déroulement des différentes tâches avec les meilleures conditions de travail et assure la conformité des informations entre les services, elle englobe les services contrôle de gestion informatique MQ/HSE, RH comptabilité approvisionnement.

1.8.2. Département secrétariat de direction

Comprend deux postes, une secrétaire assistance et une standardiste, ses principales missions sont l'assistant et d'assurer l'accueil physique, de répondre aux appels, aux courriers et aux emails.

1.8.3. Département informatique

Le département informatique a pour objectif d'assurer le développement de l'entreprise et de sélectionner les logiciels pour la gestion de l'entreprise. Il est également chargé d'établir la politique informatique et de garantir la sécurité et la disponibilité des données de l'entreprise.

1.8.4. Département RH

Le service RH a pour objectif d'apporter à l'entreprise le personnel nécessaire à son bon fonctionnement. Il se compose d'un chef ou d'un responsable de service, d'un chargé social, d'un chargé de formation, d'un suivi de paye et d'un suivi de carrière.

1.8.5. Département facturation et comptabilité

Le département de facturation et comptabilité compte deux personnes qui se chargent respectivement de la facturation et des tâches comptables et financières. Leur rôle principal est d'assurer la conformité des opérations comptables, d'établir facture et d'enregistrer les paiements des clients, tout en s'occupant des achats de fournitures nécessaires.

1.8.6. Département commercial

Le département commercial est crucial pour le succès de l'entreprise. Il est chargé de la collaboration entre différents départements et de la réalisation des objectifs de l'entreprise.

1.8.7. Département d'exploitation

Le département d'exploitation chez BL comprend un responsable exploitation TRM un chargé de suivi sinistre, un chargé des dossier d'exploitation et un coordinateur TRM. Chacun de ces postes joue un rôle spécifique dans la gestion des opérations et la coordination des activités liées au transport routier de marchandises.

1.9. Diagramme réseau informatique de BL

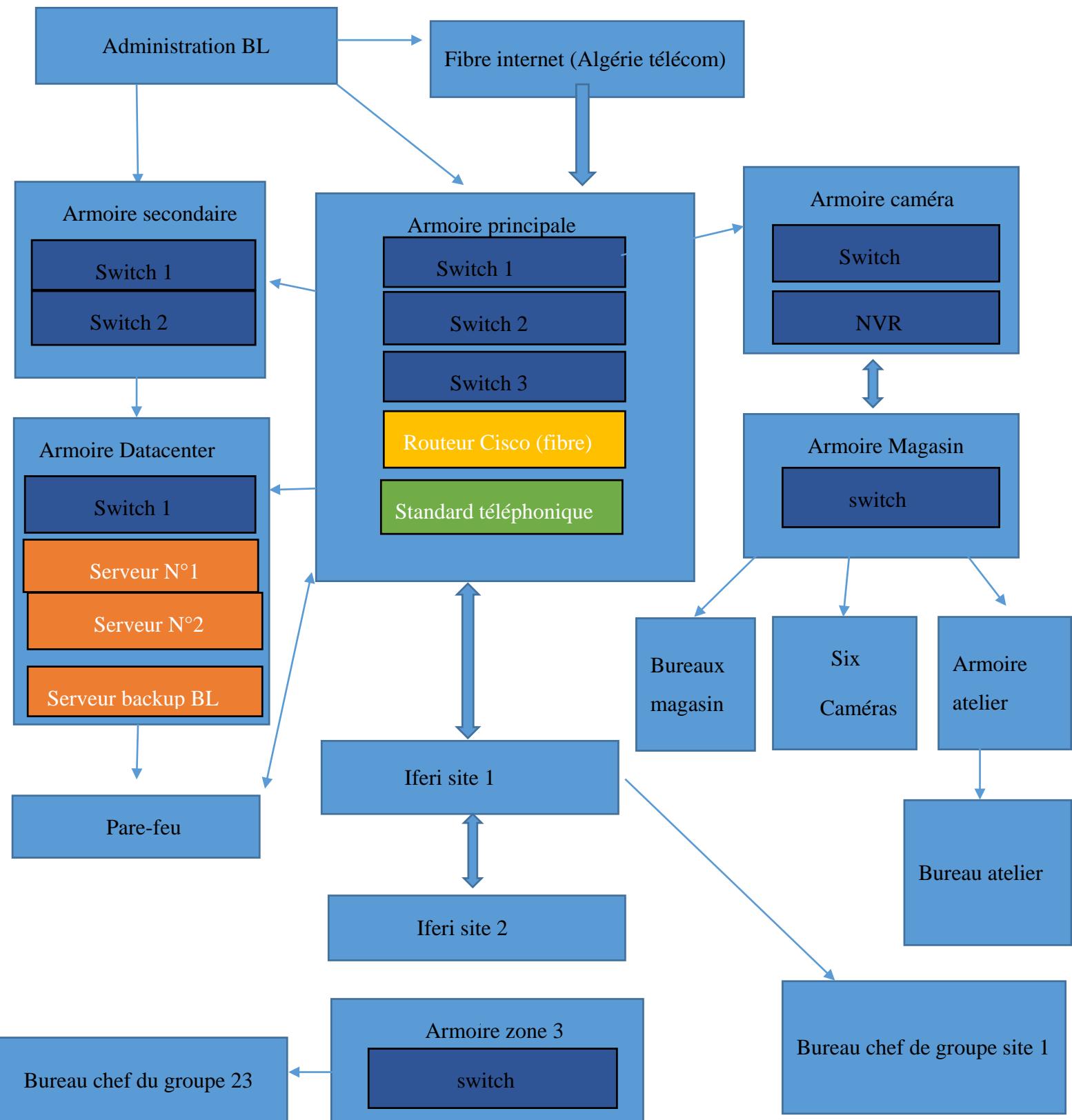


Figure 4– Architecture de réseaux de BL

1.10. Explication détaillée du diagramme réseau

Le diagramme montre un réseau informatique complexe composé de plusieurs éléments, notamment des commutateurs, des routeurs, des serveurs, des caméras et des téléphones. Le réseau est divisé en plusieurs zones, chacune avec ses propres commutateurs et routeurs. Les zones sont reliées entre elles par des liaisons fibre optique.

Légende

- ✓ Armoire principale : C'est l'armoire principale du réseau, où se trouvent les principaux commutateurs et routeurs.
- ✓ Armoire secondaire : C'est une armoire secondaire du réseau, où se trouvent des commutateurs et des routeurs pour une zone spécifique.
- ✓ Switch : C'est un périphérique réseau qui permet de connecter plusieurs appareils entre eux
- ✓ NVR : C'est un enregistreur vidéo en réseau, qui permet de stocker et de visualiser les images des caméras de surveillance.
- ✓ Serveur : C'est un ordinateur qui fournit des services aux autres appareils du réseau, tels que le stockage de fichiers, le partage d'imprimantes ou l'accès à Internet.
- ✓ Caméras : Ce sont les caméras de surveillance qui permettent de filmer les locaux.
- ✓ Téléphone : Ce sont les téléphones IP qui permettent de passer des appels téléphoniques sur le réseau.
- ✓ Fibre optique : C'est un type de câble qui utilise la lumière pour transmettre des données.

1.11. Fonctionnement du réseau

Les appareils du réseau sont connectés les uns aux autres par des câbles Ethernet ou des liaisons fibre optique. Les commutateurs permettent de connecter plusieurs appareils entre eux sur la même zone. Les routeurs permettent de connecter plusieurs zones entre elles et de diriger le trafic vers la bonne destination. Les serveurs fournissent des services aux autres appareils du réseau. Les caméras de surveillance permettent de filmer les locaux et les images sont stockées sur un NVR. Les téléphones IP permettent de passer des appels téléphoniques sur le réseau. Zones du réseau Le réseau est divisé en plusieurs zones, chacune avec ses propres commutateurs et routeurs. Cela permet de segmenter le réseau et de mieux le contrôler.

Chapitre1.Etude de l'existant

Les zones sont reliées entre elles par des liaisons fibre optique :

- ✓ **Zone 1 :** C'est la zone principale du réseau, où se trouvent les serveurs, les bureaux et les ateliers.
- ✓ **Zone 3 :** C'est une zone secondaire du réseau, où se trouvent les bureaux du chef de groupe 23 et les caméras de surveillance.
- ✓ **Administration BL :** L'administration BL est une entité qui gère le réseau. Elle est responsable de la configuration, de la maintenance et de la sécurité du réseau.
- ✓ **Fibre Latest Algérie :** Fibre Latest Algérie est un fournisseur de services Internet qui fournit la connectivité Internet au réseau.
- ✓ **Armoire Magasin :** L'armoire magasin est une armoire secondaire du réseau, où se trouvent des commutateurs pour les bureaux du magasin.
- ✓ **Armoire Datacenter :** L'armoire Datacenter est une armoire secondaire du réseau, où se trouvent les serveurs.
- ✓ **Armoire Atelier :** L'armoire atelier est une armoire secondaire du réseau, où se trouvent des commutateurs pour les ateliers.
- ✓ **Bureaux :** Les bureaux sont les espaces de travail des employés.
- ✓ **Magasin :** Le magasin est l'endroit où sont stockés les produits.
- ✓ **Paceteu :** Paceteu est un appareil qui permet de contrôler l'accès au réseau.
- ✓ **Bureau Chef de groupe :** Le bureau du chef de groupe est l'espace de travail du chef de groupe.
- ✓ **Sitel :** Sitel est un fournisseur de services de centre d'appels.
- ✓ **IFRI SITE1 :** IFRI SITE1 est une entité qui utilise le réseau.
- ✓ **IFRI ZONE3 :** IFRI ZONE3 est une entité qui utilise le réseau dans la zone 3.
- ✓ **Bureau Chef :** Le bureau du chef de groupe 23 est l'espace de travail du chef de groupe 23.

1.12. Architectures du réseau informatique de GROUPE IFRI

L'architecture réseau actuelle d'Ifri est une architecture réseau traditionnelle, avec une gestion manuelle. Cela signifie que les décisions de routage et de commutation sont prises par les périphériques réseau eux-mêmes, et que la configuration et la gestion du réseau sont effectuées manuellement par les administrateurs réseau. L'architecture réseau actuelle de groupe Ifri est repartie sur quatre sites reliés par la fibre optique, et ces composants principaux sont les suivants :

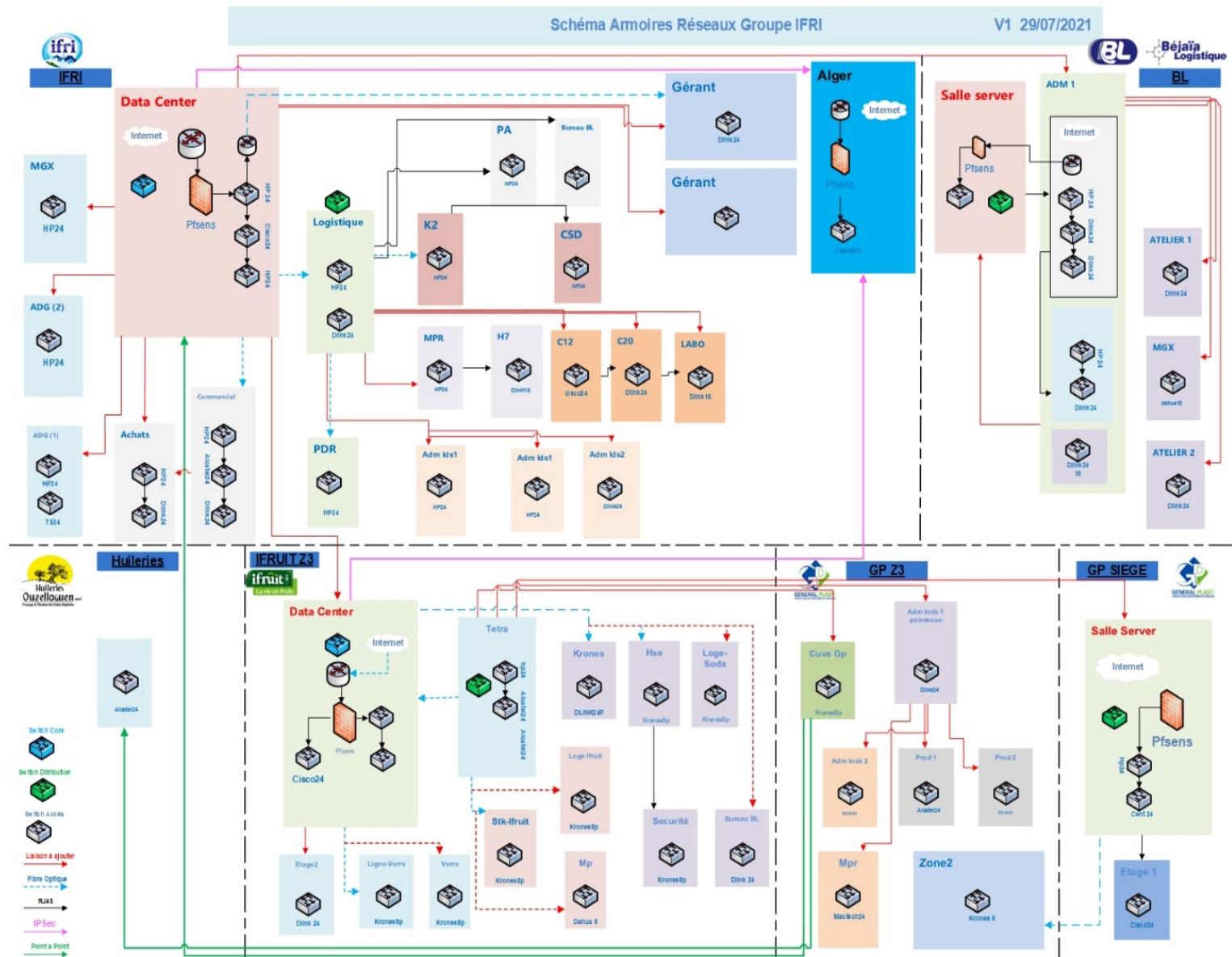


Figure 5- Architecture actuel du réseau informatique de GROUPE IFRI.

1.13. Description de l'architecture

- **Data Center** : Le data center héberge les serveurs critiques et les applications sensibles de l'entreprise. Il est connecté au reste du réseau via des routeurs et des pare-feu.
- **Salle Serveur ADM 1** : Cette salle serveur héberge les serveurs administratifs de l'entreprise. Elle est connectée au data center et au reste du réseau via des commutateurs et des routeurs.
- **Atelier 1 et Atelier 2** : Ces ateliers hébergent les stations de travail des employés. Ils sont connectés au reste du réseau via des commutateurs et des routeurs.
- **Salle Server** : Cette salle serveur héberge les serveurs des filiales d'Ifri. Elle est connectée au reste du réseau via des routeurs et des pare-feu.
- **Pfsens** : Pfsens est un pare-feu open source qui est utilisé pour protéger le réseau contre les accès non autorisés.
- **Cisco24** : Cisco24 est un routeur Cisco qui est utilisé pour connecter le réseau d'Ifri à Internet.
- **Mp** : Mp est un pare-feu Palo Alto Networks qui est utilisé pour protéger le réseau contre des attaques.
- **Mpr** : Mpr est un routeur Cisco qui est utilisé pour connecter le réseau d'Ifri au réseau du fournisseur d'accès Internet (FAI).
- **Zone2** : La zone 2 est une zone de sécurité qui héberge les serveurs et les applications critiques d'Ifri. Elle est isolée du reste du réseau pour des raisons de sécurité.

Elle est constituée de plusieurs équipements, des commutateurs et des routeurs, des pc, des pare-feu, de marque Cisco.



Figure 6- D-LINK DIR-685 Wireless AC Routeur



Figure 7- Cisco Catalyst 9120axi



Figure 8- PC standard

1.14. Découverte du Serveur Central et de l'Infrastructure Réseau de l'Entreprise

Lors de notre stage pratique, nous avons eu l'opportunité de commencer par une visite approfondie du serveur central de l'entreprise IFERI. Nous avons visité une "salle serveurs", plus spécifiquement un "data center en armoire" ou une "armoire réseau climatisée". Ces armoires sont souvent utilisées pour abriter des équipements informatiques tels que des serveurs, des routeurs, des switches, et d'autres matériels réseau, tout en maintenant une température contrôlée pour assurer leur bon fonctionnement. Ce serveur joue un rôle crucial en gérant l'ensemble du réseau de l'entreprise, assurant ainsi la fluidité des opérations et la communication interne. Cette immersion initiale nous a permis de comprendre les bases de l'architecture réseau existante acquise lors de cette première phase a été fondamentale pour les étapes suivantes de notre travail.

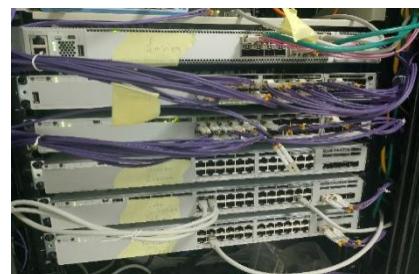


Figure 9- Centre Data IFRI

1.15. Critique sur l'architecture Actuelle du Groupe IFRI

En cherchant à sécuriser l'architecture du système, nous avons alourdi le système avec des équipements supplémentaires, ce qui a conduit à la duplication de l'activité des transactions. Cela a engendré des coûts d'achat et de maintenance trop élevés pour l'entreprise. Toutefois, il est important de penser non seulement à la sécurité, mais aussi à l'automatisation et à la centralisation offertes par l'architecture réseau SDN (Software-Defined Networking). De plus, il est crucial de protéger les données grâce à une sauvegarde automatique périodique et sécurisée. Il est donc nécessaire de critiquer cette approche avant de proposer des améliorations via une architecture réseau SDN :

- ❖ Réseau Plat : Absence de SDN.
- ❖ Absence de Segmentation Dynamique.
- ❖ Transmission de Données Non-Sécurisée.
- ❖ Accès Internet Non-Optimisé.
- ❖ Limite d'innovation.
- ❖ Absence de Configuration Automatique en Cas de Changement.
- ❖ Gestion de Réseau Local Inefficace.

1.16. Étude de l'Architectures Réseau de Bejaia Logistique

Après avoir pris connaissance du serveur central, notre travail s'est concentré sur l'analyse de l'architecture réseau de Bejaia Logistique. Cette partie de l'entreprise représente un point stratégique pour la mise en place d'une nouvelle architecture réseau SDN. L'objectif était de comprendre les défis actuels posés par l'architecture traditionnelle et de préparer le terrain pour une transition efficace vers une solution plus moderne et agile. Notre étude a révélé des faiblesses structurelles et des coûts élevés associés aux équipements réseau traditionnels, soulignant la nécessité d'une mise à jour technologique.

1.17. Architectures Réseau de BL

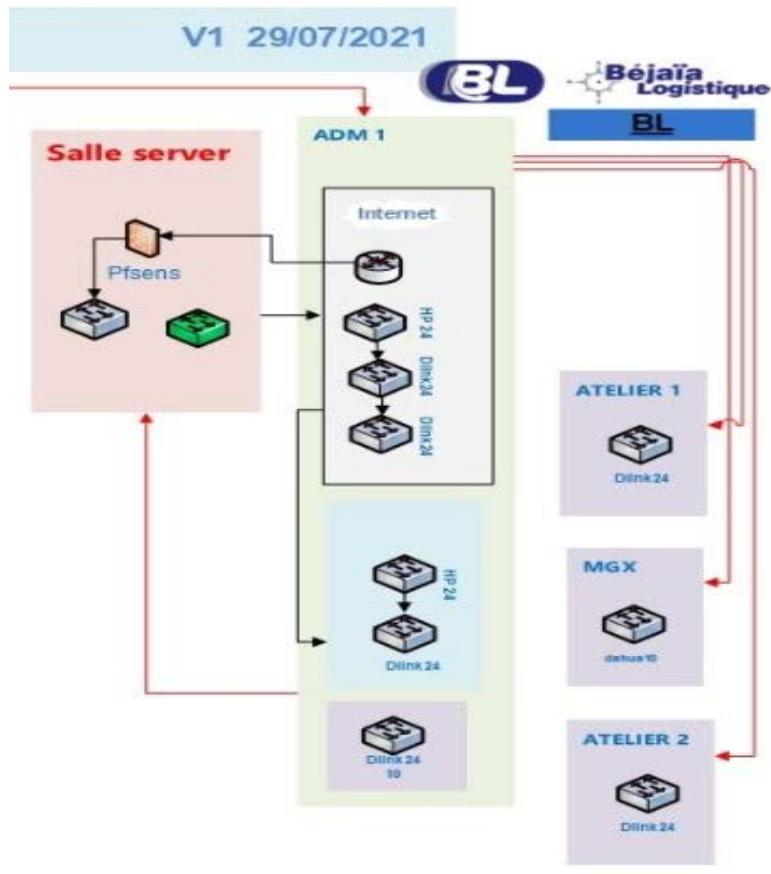


Figure 10- Architecture actuel du réseau informatique BL

1.18. Caractéristiques des équipements utilisés dans le réseau de BL

Marque de l'équipement	Caractéristique	Nombre de port	Débit	Type de port	Quantité	Durée de vie
CISCO 890 série	Routeur gigabit,vpn intégré, pare-feu	10	1000Mps	RJ45	1	5 ans
HPE 1820	Switch géré, vlan,Qos,Link aggregation	24	1000Mbps	RJ45	2	5ans
D-LINK DES 1024 D	Switch non manageable,store-and-forward	24	100Mbps	RJ45	5	5ans
D-LINK DES 1024 D/1000	Switch non manageable,gigabit ethernet	24	1000Mbps	RJ45	3	5ans

Dahua switch poe	Switch Poe pour les caméra , 150x poe	8	1000Mbps	RJ45	1	3ans
HP PRO 15 ,8GO	I5, 8GO RAM, 1TO DISQUE DUR	5	4 sur 1000 Mbps et 1 sur 100 Mbps	RJ45	4	5ans
HP PROLIANT DL380 G8	VXEON® E52620,48 GO RAM, 2 TO DISQUE DUR(en RAD)	2	1000Mbps	RJ45	2	5ans
PC standard	17.25GO RAM, 1.5TO DISQUE SUR	1	100Mbps	RJ45	8	5ans
D-LINK DIR-685	Routeur sans fil AC.double bande,ports Gigabit, NAS,LCD	4	1000Mbps	RJ45/Wi-fi	1	3ans
Cisco castalyst 9120axi	Point d'accés wi-fi 6, MU-MIMO,OFDMA ,antennes, Internet	1	4800Mbps	Wi-fi	2	5ans

Tableau 2- caractéristiques des équipements utilisés dans le réseau de BL

1.19. Problématique

L'architecture réseau traditionnelle a longtemps été caractérisée par une complexité et une difficulté de gestion manuelle, notamment en raison du câblage complexe et étendu nécessaire pour connecter les différents équipements.

Bejaia Logistique (BL), une entreprise de transport routier en Algérie, cherche à moderniser son infrastructure réseau pour rester compétitive. Face aux défis actuels, tels que :

Les administrateurs doivent souvent effectuer des configurations manuelles complexes, ce qui augmente le risque d'erreurs humaines et de problèmes de configuration.

L'absence d'automatisation efficace oblige les administrateurs à déployer et gérer les politiques de sécurité de manière individuelle sur chaque dispositif réseau, ce qui est chronophage et sujet à des erreurs.

L'agrandissement de l'architecture réseau et la diversité des équipements réseau nécessite plus de matérielles et logicielles qui peuvent entraver la croissance du réseau ainsi le besoin des outils de supervision et de gestion spécifiques pour chaque type.

Le cout accru en termes de temps et de ressources pour surveiller et gérer un réseau complexe

Les équipements standards sont moins économiques que les équipements modernes.

Enfin la transition vers le SDN chez BL nécessite une planification minutieuse, incluant l'audit de l'infrastructure actuelle, la rationalisation du câblage existant pour s'adapter aux nouvelles exigences, la formation du personnel et, si nécessaire, le recrutement de spécialistes en SDN. Une gestion efficace du changement est cruciale, impliquant une communication claire avec toutes les parties prenantes et un plan de transition détaillé pour minimiser les perturbations. En résumé, bien que la transition vers un réseau SDN présente des défis, elle offre des opportunités significatives pour améliorer la sécurité, l'efficacité et la flexibilité du réseau de BL, tout en simplifiant le câblage et la gestion de l'infrastructure physique.

1.20. Les Solutions proposée

1.20.1. Présentation de la première solution

Voici une vue globale de la nouvelle architecture proposée au groupe IFRI : L'architecture que nous avons proposée s'est basé sur le me modèle hiérarchique en trois couche qui se compose en trois parties principales :

- Couche cœur
- Couche distribution
- Couche accès

Cette architecture Caractéristiques Structure en couches : divisée en trois couches (accès, distribution, cœur). Commutateurs et routeurs fixes : Utilisation de matériel dédié pour chaque couche avec des rôles bien définis.

Routage statique ou dynamique : Les protocoles de routage sont configurés pour diriger le trafic.

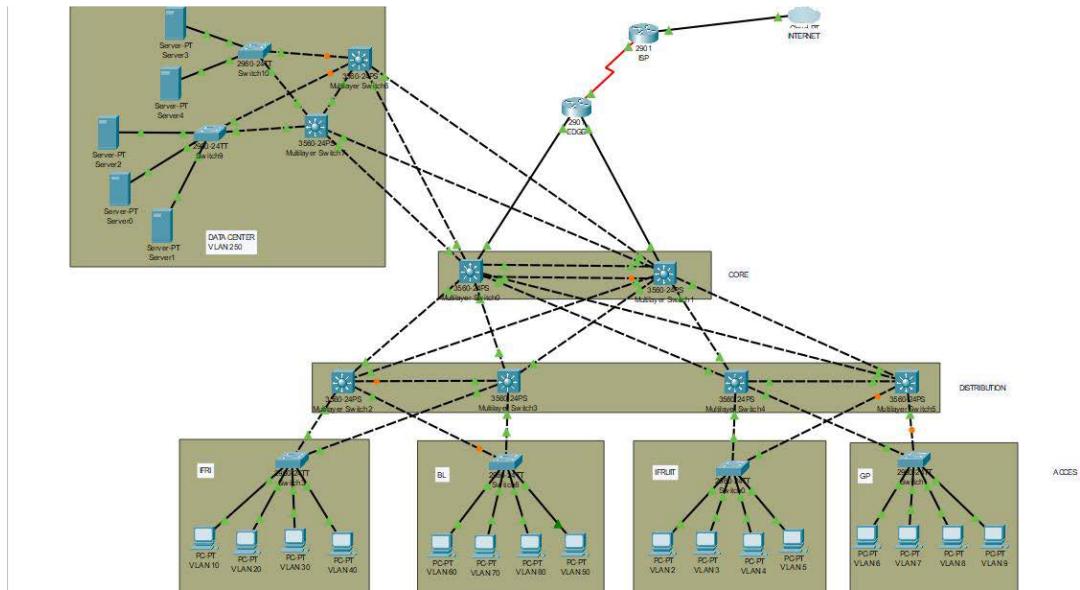


Figure 11- Architecture proposée 1

1.20.2. Inconvénients de l'architecture réseau proposée 1

L'architecture hiérarchique traditionnelle présente une rigidité notable, ce qui la rend difficile à adapter rapidement aux changements de la demande ou à l'intégration de nouvelles technologies. De plus, les coûts d'expansion peuvent être élevés, car l'ajout de nouvelles capacités ou fonctionnalités nécessite souvent des investissements matériels significatifs. Enfin, la gestion de ce type de réseau peut devenir complexe et laborieuse à mesure que sa taille et sa complexité augmentent, rendant le dépannage et la maintenance plus difficiles.

1.20.3. Présentation de deuxième solution

L'architecture réseau présentée est une topologie SD-ACCESS est une solution très moderne, mais elle nécessite des équipements coûteux, rendant son adoption difficile pour certaines entreprises. De plus, sa mise en œuvre requiert une expertise technique spécialisée, ce qui peut représenter un obstacle supplémentaire

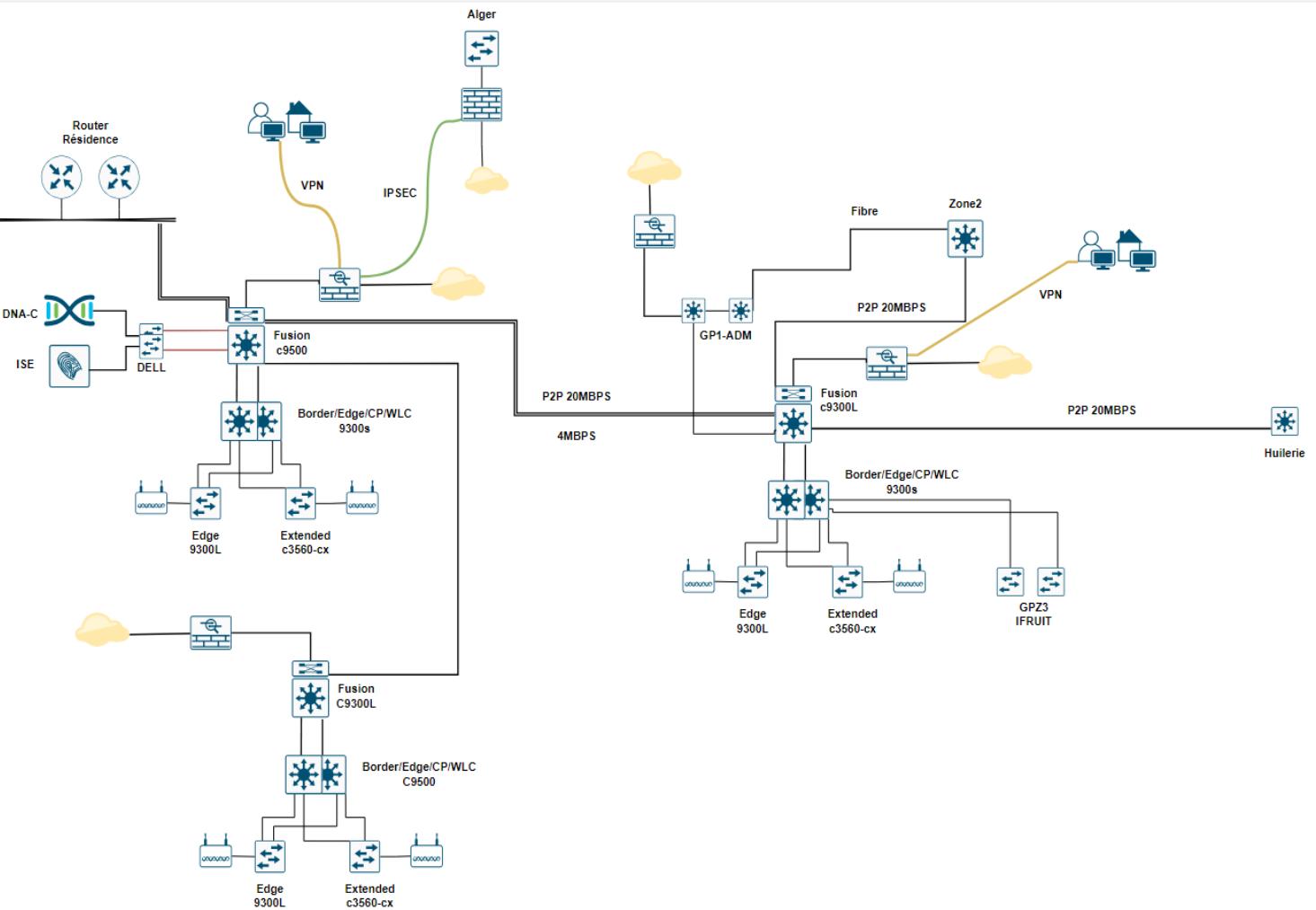


Figure 12- Architecture proposée 2

1.20.4. Inconvénients de l'architecture réseau proposée 2

- ✓ SD-ACCESS nécessite souvent l'achat de nouveaux équipements compatibles avec les technologies Cisco DNA, ce qui peut représenter un investissement substantiel.
- ✓ Le coût des licences pour les logiciels Cisco DNA Center et autres composants logiciels peut être prohibitif pour certaines entreprises.

- ✓ La mise en œuvre de SD-ACCESS nécessite une expertise technique spécialisée et une formation approfondie pour le personnel IT, ce qui peut être un obstacle pour certaines entreprises.
- ✓ L'intégration avec les infrastructures existantes peut être complexe et nécessite une planification minutieuse et des ressources dédiées.
- ✓ Adopter l'architecture SD-ACCESS signifie devenir fortement dépendant de Cisco pour le matériel, le logiciel et le support technique, limitant ainsi la flexibilité de l'entreprise à choisir d'autres solutions.
- ✓ Les entreprises sont contraintes de suivre l'évolution des produits et services de Cisco, ce qui peut entraîner des coûts récurrents pour les mises à jour et les nouvelles versions.

1.20.5. Présentation de la troisième solution

Le réseau IP prend du temps pour la configuration et le dépannage car il nécessite l'accès à chaque interface de ligne de commande (CLI) ou interface utilisateur graphique (GUI) de l'appareil. En d'autres termes, le plan de contrôle (collecte d'informations pour transmettre des données) et le plan de données (plan de transfert de données) sont exécutés sur chaque périphérique intermédiaire.

Nous proposons une architecture simplifiée, juste le nécessaire, facile à maintenir et à gérer, efficace, optimale, prend en charge toute l'activité de flexibilité, gestion centralisée, et amélioration de la sécurité.

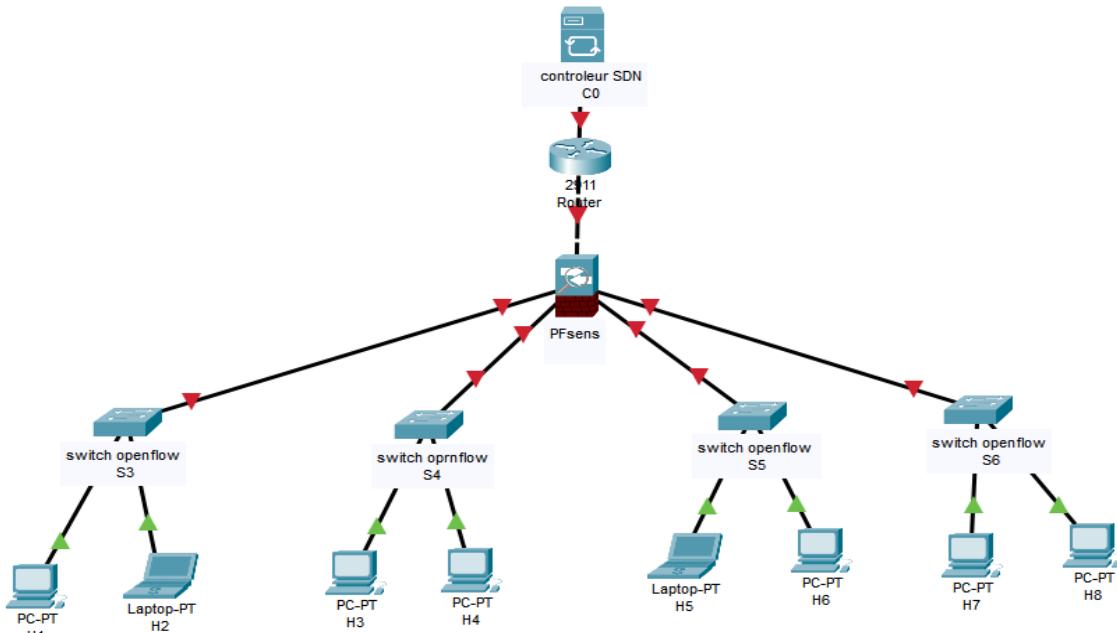


Figure 13- Architecture proposée 3

1.21. Solution retenue

En tant que stagiaire chez Bejaia Logistique, nous avons étudié différents scénarios et analysé les besoins de l'entreprise en matière d'amélioration de la sécurité des données et du contrôle d'accès aux ressources en ligne. Après notre analyse, nous avons proposé la mise en place d'un réseau SDN comme solution adaptée à ces besoins. Cette solution permet à Bejaia Logistique de moderniser son infrastructure réseau, offrant ainsi une meilleure sécurisation des données sensibles en empêchant les accès non autorisés et en cryptant les échanges de données. De plus, elle permet à l'entreprise de contrôler de manière granulaire l'accès à ses ressources en ligne, en fonction des besoins de chaque utilisateur, tout en optimisant l'utilisation de la bande passante. Grâce à la mise en place d'un réseau SDN, Bejaia Logistique pourra centraliser et automatiser la gestion de son réseau, simplifiant ainsi la gestion des comptes utilisateurs et permettant une mise à jour en temps réel des informations d'identification. De plus, les administrateurs auront la possibilité de définir des règles précises pour autoriser ou bloquer l'accès à des sites spécifiques en fonction des politiques de l'entreprise, renforçant ainsi la sécurité et la conformité aux normes réglementaires.

1.22. Objectif et résultat visés

- Mettre en place une nouvelle architecture réseau basée sur la technologie SDN.
 - Assurer un routage optimisé et flexible grâce à l'utilisation de protocoles et technologies adaptés au SDN.
 - Mettre en place des mécanismes de sécurité avancés et adaptés aux environnements SDN.

1.23. Conclusion

En conclusion, la mise en place d'un réseau SDN chez Bejaia Logistique représente une étape cruciale pour répondre aux besoins croissants de l'entreprise en matière de sécurité des données et de flexibilité opérationnelle. En modernisant son infrastructure réseau traditionnelle, Bejaia Logistique pourra relever les défis actuels liés à la gestion des accès, à la sécurisation des données sensibles et à la gestion de la bande passante. La transition vers un réseau SDN offre à Bejaia Logistique l'opportunité de renforcer la protection de ses données contre les attaques externes et internes, tout en garantissant un contrôle précis sur les accès aux ressources en ligne. De plus, la

CHAPITRE1. ETUDE DE L'EXISTANT

centralisation de la gestion du réseau et la flexibilité offerte par une architecture SDN permettront à l'entreprise de s'adapter plus facilement aux évolutions futures de ses besoins informatiques. En mettant en œuvre un réseau SDN, Bejaia Logistique démontre son engagement envers la modernisation de ses infrastructures technologiques pour répondre aux exigences croissantes du marché et offrir à ses employés et clients un environnement de travail plus sécurisé et efficient .

Chapitre 2

Réseaux défini par logiciel SDN

2.1. Introduction

Un réseau est le résultat d'interconnexion entre plusieurs machines afin que les utilisateurs et les applications qui y sont exécutées puissent échanger des informations.

Le contrôle distribué et les protocoles de réseau de transport en cours d'exécution à l'intérieur des routeurs et des commutateurs sont les technologies clés qui permettent aux informations, sous la forme de paquets numériques, de voyager dans le monde entier. Malgré leur adoption généralisée, les réseaux IP traditionnels sont complexes et difficiles à gérer. Depuis 2008, on assiste à une nouvelle tendance forte à la mise en réseau avec les réseaux SDN. L'émergence de cette nouvelle tendance dans l'architecture réseau est motivée par la volonté de mettre en œuvre des principes de conception dans le domaine du réseau.

Dans ce chapitre nous discuterons en premier lieu sur les différentes problématiques des réseaux traditionnels, et les principes du réseau SDN et ces avantages. Et les différentes couches.

2.2. Principe de réseau SDN

2.2.1. Réseau traditionnelle vers réseau SDN

Les équipements réseaux traditionnels utilisés actuellement comportent deux parties fondamentales : plan de contrôle et plan de données (le Cerveau et le corps).

La fonction de la première partie se résume en la prise de décision et le lancement des processus, comme le routage ou redirection du trafic, par contre la deuxième partie traite la mise en œuvre de toutes les décisionsprises par la première partie, c'est-à-dire lorsqu'un paquet arrive sur un port d'un commutateur ou d'un routeur, celui-ci applique les règles de routage ou de commutation qui sont inscrites dans son système d'exploitation. Il est difficile de gérer les réseaux et d'appliquer efficacement de nouvelles politiques, car chaque périphérique fonctionne.

Avec un protocole d'un niveau donné et doit être configuré de manière appropriée pour communiquer les uns avec les autres.

Donc il n'est pas facile d'ajouter une règle parce qu'il n'y a pas de moyen standardisé pour le faire, mais aussi parce qu'un périphérique risqued'entrer en conflit avec l'un des nombreux autres périphériques du réseau. En outre, les règles étaient initialement destinées à être statiques et donc le réseau est statique et mal adapté aux besoins et technologies actuels.

La vie d'un administrateur réseau est un cycle sans fin de réflexion, de travail fastidieux, de configurations, de mises à niveau et de remplacements. Pour les entreprises et les opérateurs, c'est une activité coûteuse. La mise en réseau n'a pas connu l'équivalent de la révolution des PC (*Personale Computer*) tel que banaliser le matériel, offrir un choix entre plusieurs systèmes d'exploitation et créer un marché d'applications concurrentes.

Tous les fabricants des machines réseau ont développé des systèmes très compliqués et très incompatibles. [17]

2.2.2. Problématiques des réseaux traditionnels

Ces réseaux traditionnels sont désavantagés par les points suivants :

- **Complexité** : l'ajout ou la modification d'équipements et l'implémentation des politiques réseaux sont complexes, longues et peuvent être source d'interruptions de service. Cela décourage les modifications et l'évolution du réseau.
- **Passage à l'échelle** : l'impossibilité d'avoir un réseau qui s'adapte au trafic a obligé les opérateurs à sur-provisionner leurs réseaux ce qui ajoute une complexité de gestion sur le plan de contrôle.
- **Dépendance aux constructeurs** : les constructeurs réalisent des produits avec des durées de vie et un manque de standard, d'interface ouverte. Ce qui restreint les opérateurs réseaux d'adapter le réseau à leurs propres besoins. Une telle architecture statique est inadaptée aux besoins de stockages dynamiques des centres de données, des campus et environnements des prestataires de services réseaux d'aujourd'hui.[18]

Ce qui cause un très large impact et conséquences qui sont mis en évidence dans le schéma suivant :

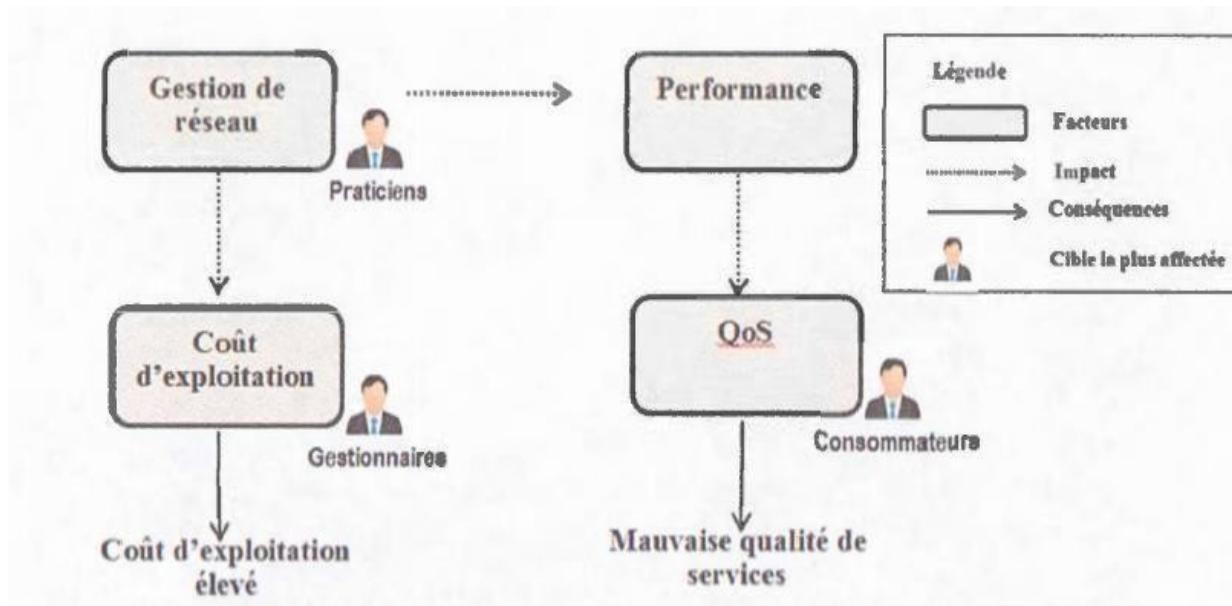


Figure 14- Impacts et conséquences de l'architecture réseau traditionnel. [18]

La figure montre que la gestion des réseaux traditionnels d'aujourd'hui a une influence négative sur une grande partie de la population (praticiens, gestionnaires et consommateurs des services réseaux). Les conséquences qui en découlent sont, la mauvaise qualité de service pour les consommateurs, et les coûts d'exploitation élevés pour les gestionnaires. Il est alors extrêmement important d'étudier ce problème afin de palier à cette situation, ce qui représente d'énormes défis.

2.2.3. La comparaison entre réseau traditionnel et SDN

	Réseau SDN	Réseau traditionnel
Fonctionnalité	<ul style="list-style-type: none"> -Découpe le plan de contrôle de celui du plan de données. -Offre un meilleur contrôle du réseau et la possibilité de le programmer. 	-Le contrôle du réseau est complexe.

CHAPITRE2. RESEAU DEFINIT PAR LOGICIEL SDN

Configuration	- Configuration automatique à travers une centralisation du contrôle du réseau. - Optimisation de la configuration.	- Une configuration manuelle et la possibilité de faire des erreurs qui vont entraîner un comportement erroné du réseau.
Performances	- Contrôle global de l'information.	- Le problème de configuration statique
Innovation	- Implémentation facile de logiciel et des mises à jour dans le réseau. - Environnements de tests suffisants.	- Difficultés d'implémentation de logiciels et des mises à jour dans le réseau. - Environnements de tests limités.

Tableau 3- Tableau comparatif entre le SDN est les réseaux traditionnels[19]

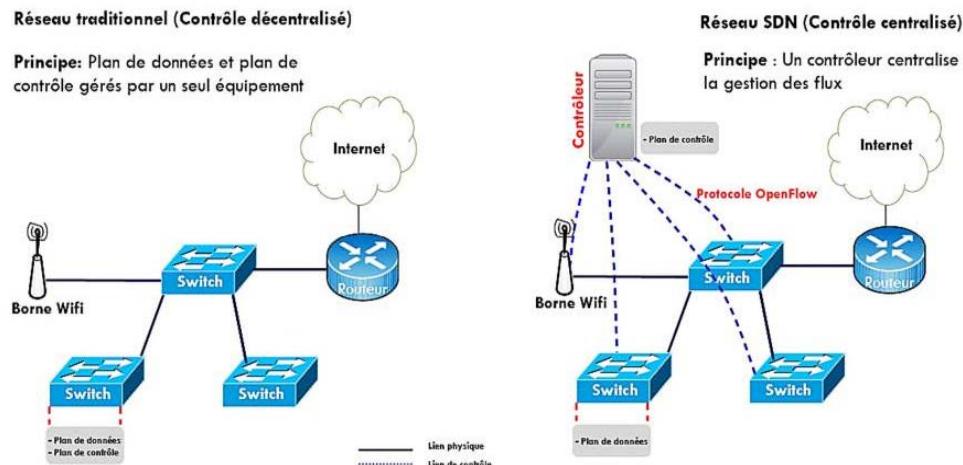


Figure 15- Comparaison entre les contrôles traditionnels et centralisés d'un SDN[20]

2.2.4. Historique

Avant l'apparition des réseaux SDN tels que nous les connaissons aujourd'hui, plusieurs idées et travaux ont été proposés auparavant, notamment la programmation du réseau et la séparation des plans de contrôle et de données. Par exemple (AN, Active Network) et (PN, Programming Network) le projet de recherche nommé DCAN (Devolved Control of ATM Networks). Le début des réseaux SDN a commencé avec le projet Ethane, lancé en 2006 à l'université de Stanford. En effet, le projet Ethane définit une nouvelle architecture pour les réseaux d'entreprises. L'objectif d'Ethane était d'avoir un contrôleur centralisé pour gérer les règles (politiques) et la sécurité dans le réseau. Ethane utilise deux composantes : un contrôleur

CHAPITRE2. RESEAU DEFINIT PAR LOGICIEL SDN

pour décider si un paquet doit être retransmis, et un switch Ethane composé de table et d'une chaîne de communication entre les deux. [21]

La figure ci-dessous montre le développement de réseau SDN dans le temps :

- 2007-2011 : l'apparition du réseau SDN.
- 2012-2013 : la maturité du réseau SDN.
- 2014-2015 : l'automatisation du réseau SDN.
- 2015-2016 : Les déploiements de SDN ont commencé à progresser.
- 2016-2019 : la généralisation du réseau SDN.
- 2020 : l'économie du SDN.

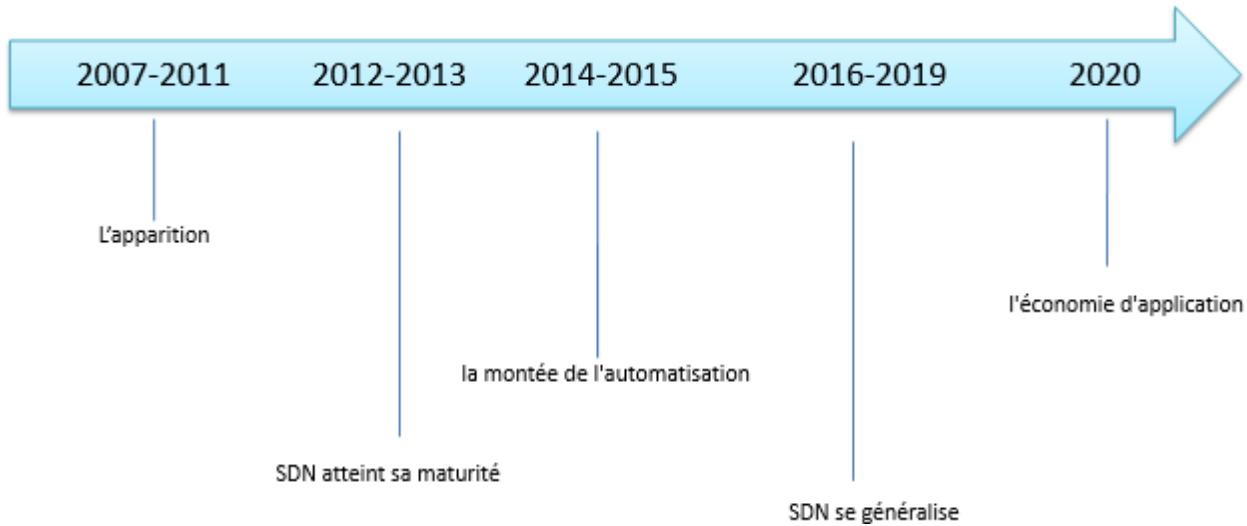


Figure 16- le développement d'un réseau SDN.[21]

Il y a une réalité SDN et cela conforte les analystes dans leurs prédictions : le métier d'administrateur réseau devrait prochainement changer. Les chiffres suivants démontrent bien que l'engouement pour le SDN n'est pas juste une mode ou un simple intérêt technologique :

Croissance des investissements depuis 2007

419 entreprises se proclament comme fournissant des solutions «SDN»

Marché estimé à 35 milliards de dollars pour 2008

2.2.5. Définition du SDN

Est nouvelle façon de concevoir les réseaux de communication, il peut utiliser et vendu par les opérateurs, a l'origine plusieurs technologies qui ne s'appliquent au même périmètre.

Il permet entre autres, de dissocier le plan donné du plan de contrôle au niveau des équipements interconnexion.

Et le SDN est une architecture émergente qui est dynamique, individuelle, rentable, adaptable, ce qui la rend idéale pour lanature dynamique et les applications à large bande passante.

Le réseau SDN est un ensemble de techniques utilisées pour faciliter la conception, la livraison, l'exploitation des servicesdes réseaux d'une manière dynamique et évolutive. [23]

2.2.6. C'est quoi SDN

Plus concrètement, on peut dire qu'une architecture réseau suit le paradigme SDN si, et seulement si, elle vérifie les points suivants :

- ✓ **Le plan de contrôle est complètement découplé du plan de données**, cette séparation est matérialisée à travers la définition d'une interface de programmation (Southbound API)
- ✓ **Toute l'intelligence du réseau est externalisée dans un point logiquement centralisé appelé contrôleur SDN**, ce dernier offre une vue globale sur toute l'infrastructure physique.
- ✓ **Le contrôleur SDN est un composant programmable** qui expose une API (Northbound API) pour spécifier des applications de contrôle. [22]

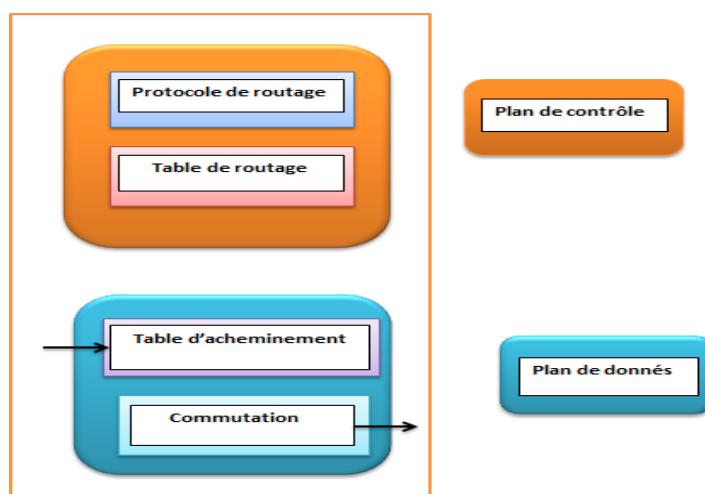


Figure 17- Architecture d'un routeur.[23]

2.2.7. Architecture de SDN

Dans cet exemple architecture SDN nous avons un contrôleur SDN centralisé qui gère un réseau de commutateur SDN. Les applications SDN interagissent avec le contrôleur via une API pour contrôler le flux de données dans le réseau. Cette architecture offre une flexibilité et une gestion simplifiée du réseau.

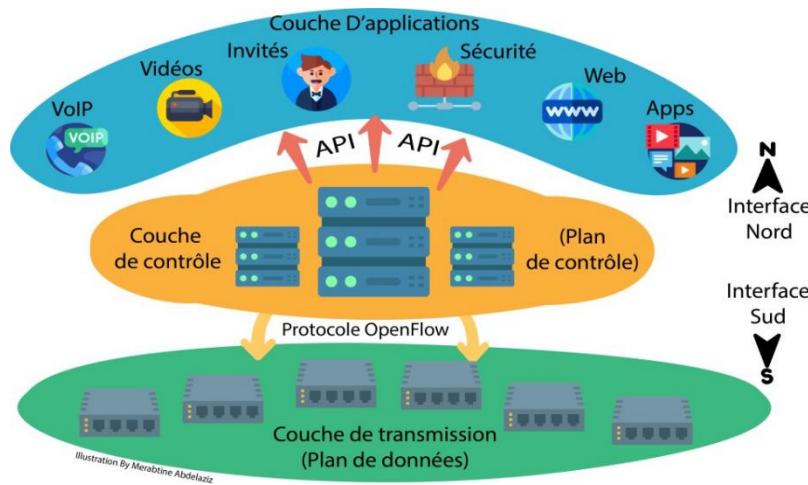


Figure 18- Architecture SDN.[23]

Comme nous le constatons, ce qui différencie un réseau SDN et un réseau traditionnel est le fait que l'infrastructure est la principale différence entre les SDN et les réseaux traditionnels, donc le SDN repose sur un logiciel, tandis que les réseaux traditionnels sont basés sur du matériel. Le plan de contrôle étant basé sur un logiciel, le SDN est beaucoup plus flexible qu'un réseau traditionnel. Il permet aux administrateurs de contrôler le réseau, modifier les paramètres de configuration, provisionner les ressources et augmenter la capacité du réseau, et ce à partir d'une interface utilisateur centralisée, sans ajouter de matériel supplémentaire. Il existe également des différences concernant la sécurité entre le SDN et le réseau traditionnel. Grâce à une visibilité accrue et à la possibilité de définir des voies sécurisées, Cependant, comme les réseaux software-defined utilisent un contrôleur centralisé, la protection du contrôleur est essentielle au maintien d'un réseau sécurisé, et ce point de défaillance unique représente une vulnérabilité potentielle du SDN. [23]

2.2.7.1. SDN se compose de 3 couches

➤ La couche infrastructure (la couche de transmission)

C'est la couche la plus basse, elle contient les équipements de transmission (FEs : Forwarding Elements) tels que les switch physiques et virtuels. Son rôle principal est l'acheminement du trafic et qui supportent le protocole OpenFlow qu'ils partagent avec le contrôleur. [34]

➤ La couche de contrôle

Ces contrôleurs utilisent des interfaces southbound pour contrôler le comportement des FEs et communiquent via des APIs Northbound avec la couche supérieure pour superviser et gérer le réseau. C.à.d. offre une visibilité globale du réseau et des équipements d'infrastructure. [34]

➤ La couche applicative

Héberge les applications qui peuvent introduire de nouvelles fonctionnalités réseau, comme la sécurité, la configuration dynamique et la gestion et apporte l'automatisation à travers du réseau, et à l'aide des interfaces programmables alternative. [34]

➤ Les interfaces de communication

SDN a introduit plusieurs types d'interfaces de programmation pour permettre à divers éléments de l'architecture SDN d'interagir les uns avec les autres. Cette section détaille ces interfaces de programmation. [24]

➤ Interfaces Sud

C'est dans ces interfaces que le contrôleur interagi avec ensembles des équipements de la couche infrastructure du réseau, et ce, via le protocole OpenFlow que nous verrons par la suite dans les sections suivantes, il existe belle et bien d'autres alternatives que ce protocole, mais il " est actuellement le standard de facto, qui est largement accepté et répandu dans les réseaux SDN. [25]

La figure ci-dessous montre le Processus de communication entre les Switch OpenFlow et le contrôleur dans ce cas nous avons pris le casd'un seul switch.

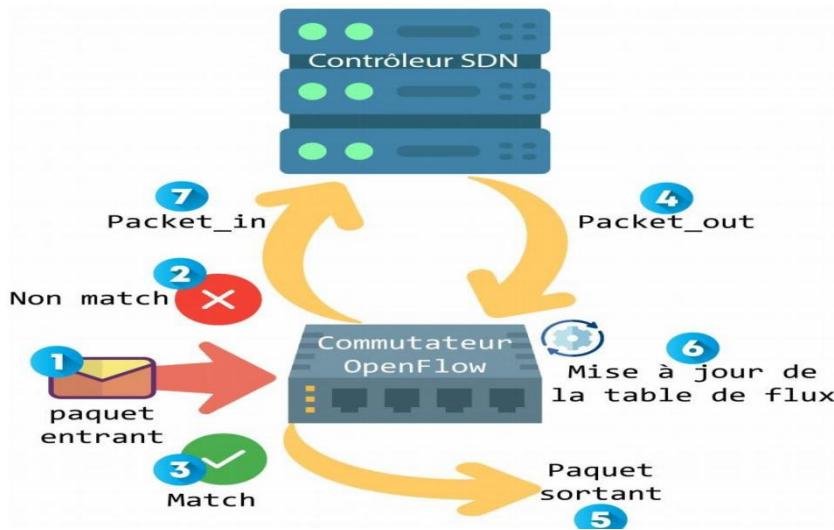


Figure 19– Processus de communication entre le Switch et le contrôleur[23]

Quand un paquet est reçu par le commutateur (1), son champ entête est examiné et comparé avec le champ Match (3) dans les entrées de la table de flux. Si le match est identifié, le commutateur exécute l'action correspondante dans la table de flux. Par contre, s'il n'y a pas de match (2), une demande est envoyée au contrôleur (7) sous la forme d'un Packet_in, puis le contrôleur décide de l'action à effectuer soit le paquet est détruit ou bien seraient orienter vers une sortie du switch (5). Selon sa configuration une action pour ce paquet, et envoie une nouvelle règle de transmission sous la forme d'un Packet_out et Flow-mod au commutateur (4). Enfin, la table de flux du commutateur est actualisée (6).

➤ Interfaces Nord

Une interface nord est une interface de programmation d'application (API) qui nous permet de communiquer avec un contrôleur, toutes les fonctions programmées du réseau, passent par cette interface, et permettent aux applications de consommer des ressources réseau et de modifier leur comportement de manière dynamique. "Le RESTful considéré comme l'API nord le plus répandue dans les réseaux SDN.[26]

➤ Interfaces Est/Ouest

Les interfaces Est/Ouest sont des interfaces de communication qui permettent la communication entre les contrôleurs dans une architecture multi-contrôleurs pour synchroniser l'état du réseau. Ces architectures sont très récentes et aucun standard de

communication inter-contrôleur n'est actuellement disponible. [27]

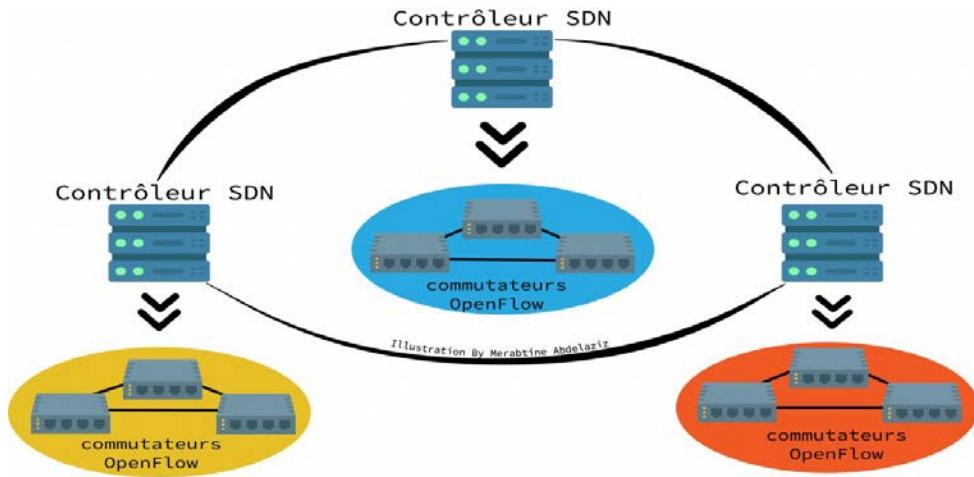


Figure 20- Communication inter-contrôleur dans une architecte distribuée[23]

2.2.8. Domaines d'application le réseau SDN

Les différents cas d'utilisation SDN :

➤ Centre de données

Le SDN permet une gestion centralisée du réseau, facilitant ainsi la configuration et l'optimisation des flux de données.

➤ Opérateurs de télécommunication

Les opérateurs de télécommunication peuvent utiliser le SDN pour la gestion et l'orchestration de leurs réseaux, offrant ainsi une flexibilité et une évolutivité accrues.

➤ Réseaux de campus

Le SDN facilite la gestion des réseaux de campus, permettant une segmentation efficace et une gestion centralisée des politiques de sécurité. [28]

2.2.9. Composants SDN

2.2.9.1. Contrôleur SDN

➤ Définition contrôleur SDN

Le contrôleur SDN est le cerveau du réseau, responsable de la gestion du plan de contrôle .il est chargé de superviser et de coordonner les opérationsdu réseau. En prenant des décisions intelligentes sur la façon dont les donnéeessont acheminées et traitées. Le contrôleur SDN communique avec les commutateurs SDN via des protocoles tels que openFlow NETCONF RESTCONF, leur fournissant des instructions pour contrôler le flux de données. [29]

➤ Fonctionnement du contrôleur SDN

Le contrôleur SDN définit les flux de données entre le plan de contrôle centralisé et les plans de données sur les routeurs et les commutateurs individuels. Pour pouvoir traverser le réseau, chaque flux doit être approuvé par le contrôleur SDN qui vérifie que la communication est autorisée dans le cadre de la politique réseau de l'entreprise. Toutes les fonctions complexes sont prises en charge par le contrôleur. Le contrôleur alimente les tables de flux. Les commutateurs gèrent les tables de flux. Sur chaque commutateur, la gestion des flux de paquets est assurée par une série de tables (**Table des flux, Table de groupe et Table de comptage**) mises en œuvre au niveau du matériel ou du firmware. À l'échelle du commutateur, un flux est une séquence de paquets qui correspond à une entrée spécifique dans une table de flux. [30]

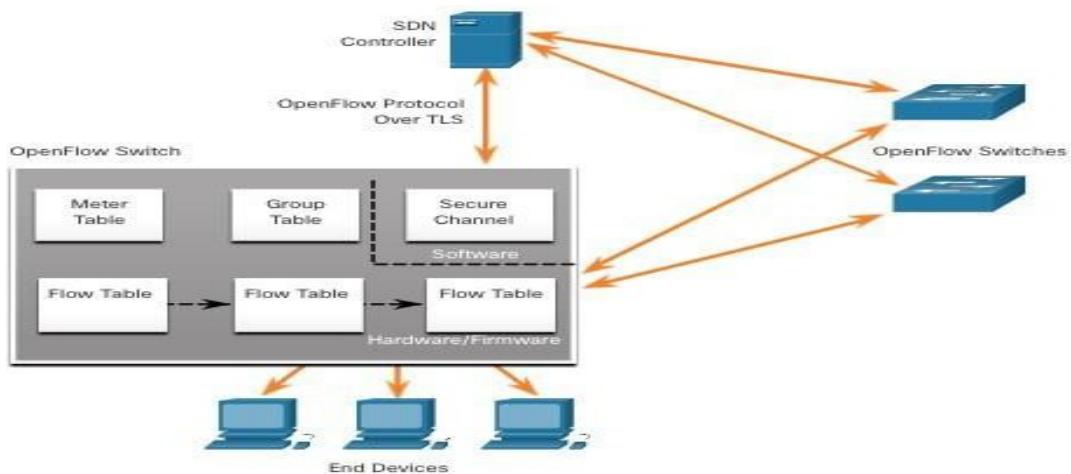


Figure 21- Fonctionnement Dun contrôleur SDN.[30]

➤ Quelque contrôleurs SDN

Il existe de nombreux contrôleurs SDN, libres et commerciaux. Les contrôleurs bien connus sont les suivants. [23]

✓ NOX

Initialement développé chez Nicira, NOX est le premier contrôleur OpenFlow, il est Open-source et ne contient que la prise en charge de C++.



Figure 22– Logo du contrôleur NOX

✓ POX

Il s'agit d'un contrôleur open source écrit en Python qui fournit un cadre pour développer et tester des contrôleurs OpenFlow comme NOX, mais les performances de POX sont nettement inférieures à celles des autres contrôleurs, donc ne convient pas au déploiement d'entreprise



Figure 23– Logo du contrôleur POX

✓ **Beacon**

Beacon est un contrôleur Java connu pour sa stabilité. Créé en 2010, il est toujours maintenu et est utilisé dans plusieurs projets de recherche. Ses performances en font une solution fiable pour une utilisation en conditions réelles. Ce contrôleur a également été utilisé dans d'autres projets tels que Floodlight ou OpenDaylight.



Figure 24– Logo du contrôleur Beacon

✓ **Floodlight s**

Floodlight est un contrôleur OpenFlow open source basé sur Java et alimenté par BigSwitch Networks. Sous licence Apache. Il est facile à installer et offre D'excellentes performances, Floodlight est plus une solution complète.



Figure 25– Logo du contrôleur Floodlight

✓ **OpenDaylight**

OpenDaylight est un projet de la Fondation Linux pris en charge par l'industrie. C'est Un framework open source pour faciliter l'accès au réseau défini par logiciel

(SDN), ODL est un logiciel basé sur Java et pris en charge par l'industrie, géré par le consortium Linux Fondation avec près de 50 entreprises membres, dont Brocade, Cisco, Citrix, Dell, Ericsson, HP, IBM, Juniper, Microsoft et Red Hat. La mission d'ODL est de créer une communauté collaborative qui partage et contribue au succès et à l'adoption du SDN.



Figure 26– Logo du contrôleur OpenDaylight

✓ **SNAC**

Utilise une application web pour gérer les règles du réseau. Un langage de définition des règles flexibles et des interfaces faciles à utiliser ont été intégrés pour configurer les équipements réseaux et contrôler leurs événements. [43]

✓ **Maestro**

Utilise la technologie du multithreading pour effectuer le parallélisme au bas niveau, en gardant un modèle simple de programmation pour les développeurs d'applications. Il atteint ses performances à travers la distribution des tâches du cœur aux threads disponibles et la minimisation de la mémoire consommée. En plus, Maestro peut traiter les demandes issues de multiples flux par une seule tâche d'exécution, ce qui augmente son efficacité. [43]

✓ **McNettle**

C'est un contrôleur SDN programmé par Nettle, qui est un DSL (Domain Specific Language) intégré dans Haskell, et permet la programmation des réseaux en utilisant OpenFlow. McNettle opère dans des serveurs multi-cœurs qui partagent leur mémoire pour atteindre une visibilité globale, une haute performance et une faible latence. [43]

✓ Orchestration

C'est une approche qui permet l'automatisation réseau, basé sur des règles et visant la coordination des composants matériels et logiciels nécessaires au fonctionnement d'une application ou d'un service. L'un des principaux objectifs de l'orchestration consiste à automatiser l'exécution des requêtes réseau et à réduire l'intervention humaine requise pour fournir une application ou un service. [31]

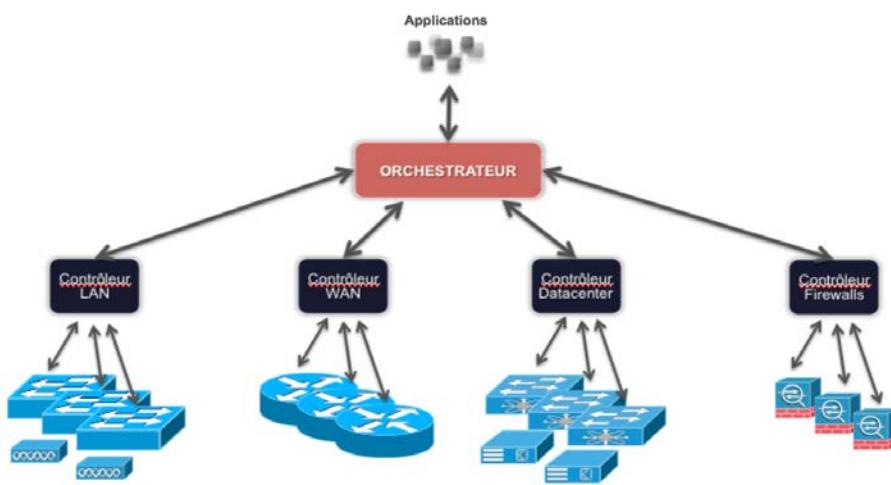


Figure 27- Architecteur orchestrateur SDN [31]

✓ Orchestration SDN

Dans un réseau SDN (Software-Defined Network) l'orchestration réseau permet à un Contrôleur SDN à travers des API de mettre à disposition, de mettre à jour et de gérer les ressources de calcul nécessaires pour fournir une application ou un service. [31]

✓ Orchestration et conteneurs

Les conteneurs représentent un nouveau domaine d'application de l'orchestration Réseau. Ils permettent d'exécuter et de déployer des applications distribuées sans utiliser de machine virtuelle dédiée. L'orchestration gère la création, la mise à niveau et la mise en service de ces conteneurs ; la technologie assure également l'interconnexion de conteneurs pour créer des applications plus complètes. Kubernetes est une plateforme open source répandue intègre des outils d'orchestration et permet de déployer et de faire fonctionner plusieurs conteneurs d'application. [31]

✓ Outils d'orchestration réseau et cas d'utilisation

L'orchestration réseau peut s'avérer utile dans plusieurs domaines :

- Automatisation de la configuration d'une interface ou routage à l'aide des protocoles IP et OpenFlow ;
- Etablissement de réseaux superposés pour diriger les plans de commande et de retransmission ; Activation de domaines de sécurité ;
- Recours à l'ingénierie du trafic pour s'assurer que les flux de travaux suivent les bons itinéraires ;
- Provisioning de services réseau, tels que l'inspection de paquets en profondeur (DPI, Deep Packet Inspection), l'inspection dynamique des paquets et les filtres, qui doivent être placés dans l'itinéraire des flux de travaux ;
- Acheminement des flux de travaux vers la destination appropriée, et marquage et gestion de ces informations. [32]

2.2.9.2. Commutateur SDN

➤ Définition de commutateur

Les commutateurs SDN sont les éléments du réseau chargés de transférer les données en fonction des instructions du contrôleur. Ils agissent comme les bras exécutants du contrôleur, acheminant les paquets de données de manière intelligente et efficace. Les commutateurs SDN reçoivent les instructions du contrôleur via des protocoles tels que OpenFlow, NETCONF et RESTCONF, et utilisent ces instructions pour prendre des décisions sur la façon de transférer les données, comme le routage et la commutation. [33]

➤ Définition Open vSwitch

Open vSwitch est un commutateur virtuel multicouche de qualité production sous licence open source Apache 2.0 . Il est conçu pour permettre une automatisation massive du réseau grâce à une extension programmatique, tout en prenant en charge les interfaces et protocoles de gestion standard De plus, il est conçu pour prendre en charge la distribution sur plusieurs serveurs physiques, similaire au vswitch distribué vNetwork de VMware, Il supporte aussi le protocole OpenFlow. [45]

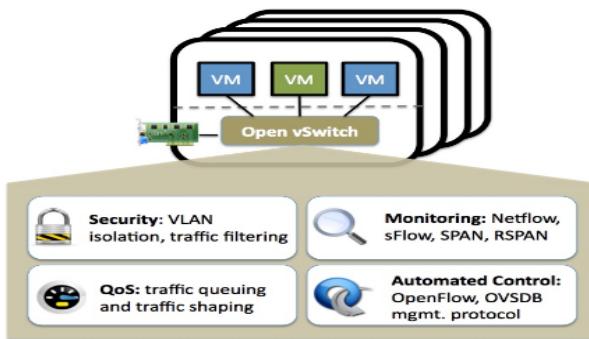


Figure 28 : open vswitch [45]

2.2.9.3. Les protocoles SDN

✓ Le protocole OpenFlow dans SDN

OpenFlow est un protocole de communication entre le contrôleur et le commutateur dans un réseau SDN. Publié par l'OpenNetworking Fondation (ONF) à Stanford. Il s'agit d'un standard ouvert utilisé par le contrôleur pour transmettre aux commutateurs des instructions qui permettent de programmer leur plan de données et d'obtenir des informations de ces commutateurs afin que le contrôleur puisse disposer d'une vue globale logique (abstraction) du réseau physique. [34]



Figure 29- logo protocole openflow.[35]

✓ La genèse d'OpenFlow

L'histoire d'OpenFlow est intéressante et permet de mieux Comprendre son rôle fondamental dans la conception de l'architecture SDN. OpenFlow a été initié comme un projet à l'université de Stanford lorsqu'un groupe de chercheurs exploraient la manière de tester de nouveaux protocoles dans

Le monde IP (créer un réseau expérimental confondu avec le réseau de production) mais sans arrêter le trafic du réseau de production lors des tests.

C'est dans cet environnement que les chercheurs à Stanford ont trouvé un moyen de séparer le trafic de recherche du trafic du réseau de production qui utilise le même réseau IP. Ils ont découvert que bien que les constructeurs de matériel réseau concevaient leurs produits différemment, tous utilisaient des tables de flux (flow table) afin d'implanter les services réseau tels que les NATs, la QoS, les firewalls, etc. Par ailleurs, bien que l'implantation des tables de flux diffère entre ces constructeurs, les chercheurs ont découvert qu'ils pouvaient exploiter un ensemble de fonctions communes.

Le résultat de l'équipe de recherche à Stanford a été OpenFlow, qui fournit un protocole ouvert (open protocol) qui permet aux administrateurs de réseau de programmer les tables de flux (flow tables) dans leurs différents routeurs IP et commutateurs Ethernet dans un but (dans le cas de Stanford) de séparer le trafic de recherche du trafic de production, chacun avec son ensemble de fonctionnalités et caractéristiques de flux. [34]

✓ Fonctionnement de OpenFlow

OpenFlow fonctionne d'une manière très simple. C'est avec un jeu d'interaction. Lorsqu'un paquet arrive à un commutateur, ce dernier vérifie s'il y a une entrée dans la table de flux qui correspond à l'en-tête du paquet. Si tel est le cas alors le commutateur exécute l'action correspondante dans sa table de flux. Et si c'est le cas contraire, qui veut dire qu'il n'y a pas une entrée correspondante, dans ce cas le commutateur envoie un message au contrôleur puis le contrôleur décide selon sa configuration une action pour le paquet et envoie une nouvelle règle de transmission au commutateur. Enfin la table de flux du commutateur est actualisée pour prendre en compte la nouvelle règle. [34]

La figure 41 suivante décrit la transmission d'un paquet avec OpenFlow.

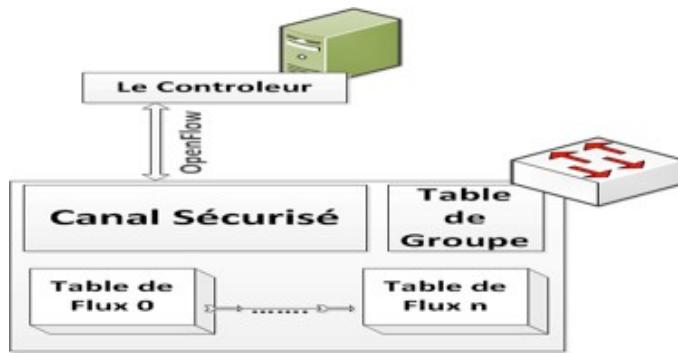


Figure 30- Architecture OpenFlow [18].

✓ **Canal OpenFlow (OpenFlow Channel)**

Le canal OpenFlow est l'interface qui connecte chaque commutateur OpenFlow à un contrôleur. Cette interface permet au contrôleur de recevoir les messages du commutateur et de pouvoir le gérer à travers le réseau. Le canal doit être sécurisé afin d'assurer le bon déroulement des communications entre le commutateur et le contrôleur. Pour cela l'échange de message se fait au cours d'une session TCP (Transmission Control Protocol) établie via le port 6653 du serveur contrôleur ou à travers une connexion SSL/TLS (Secure Sockets Layer/ Transport Layer Security). [34]

2.2.10. Établissement d'une connexion commutateur-contrôleur

Nous allons citer ci-dessous trois cas d'un établissement d'une connexion entre le commutateur OpenFlow et le contrôleur :

- **Cas n°1 : connexion établie**

Tout d'abord, il faut renseigner l'adresse IP du contrôleur au niveau du commutateur. Lors du démarrage du commutateur OpenFlow, ce dernier envoie un paquet « OFPT_HELLO » avec le numéro de version d'OpenFlow supportée. Le contrôleur vérifie la version d'OpenFlow supportée par le commutateur et lui répond par un message « OFPT_HELLO » en indiquant la version d'OpenFlow avec laquelle ils communiqueront. La connexion est ainsi établie.

La figure suivante présente les principales étapes de la connexion entre le contrôleur et le commutateur OpenFlow.

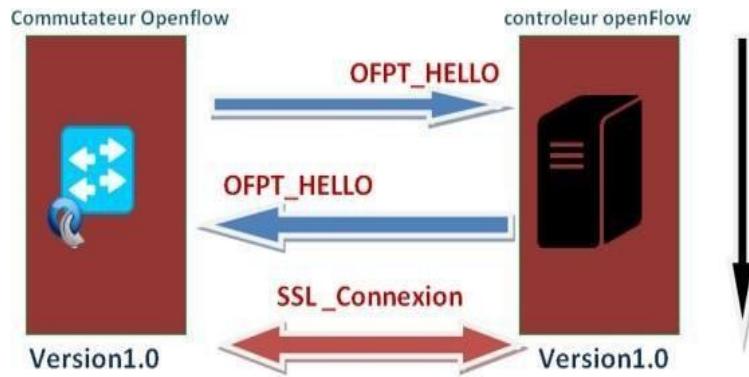


Figure 31- Connexion au contrôleur OpenFlow [36].

- **Cas n°2 : Échec de la connexion**

Comme dans le cas précédent le commutateur OpenFlow envoie un paquet « OFPT_HELLO » avec la version du protocole utilisé, le contrôleur s'aperçoit qu'il ne supporte pas la version OpenFlow du commutateur. Il lui retourne donc un paquet « OFPT_ERROR » en indiquant que c'est un problème de compatibilité, comme illustré dans la figure suivante :

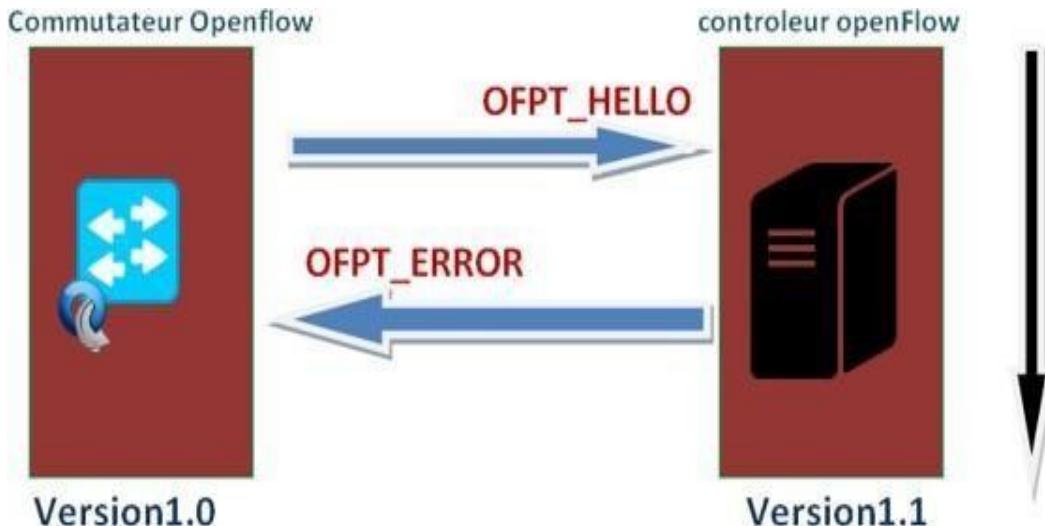


Figure 32- Échec de la connexion au contrôleur OpenFlow.[34]

- **Cas n°3 : Mode d'urgence**

Comme dans les cas précédents le commutateur envoie un paquet « OFPT_HELLO » au contrôleur, si celui-ci ne répond pas, il se met alors en mode urgence « EMERGENCY MODE ». Le commutateur utilise sa table de flux par défaut, si toutefois un paquet ne correspond à aucun enregistrement dans la table il le supprime, comme le montre la figure ci-dessous. [34]



Figure 33- Mode d'urgence.[34]

2.2.11. Processus de transmission d'un paquet avec OpenFlow

Un Switch [37] Open Flow contient une ou plusieurs tables de Flux et une Table de Groupe, qui traitent les paquets entrants et les commutent vers la destination, il s'appuie sur un Canal sécurisé pour communiquer avec le contrôleur externe. Le contrôleur gère le Switch en utilisant le protocole OpenFlow, qui lui permet d'ajouter, d'effacer et de mettre à jour les entrées dans la ou les Table de Flux. La correspondance des paquets commence par la première entrée et peut continuer ainsi, dans le cas où plusieurs Tables de Flux existent. Les entrées font la correspondance selon la priorité, si une correspondance est trouvée les instructions associées à cette entrée sont exécutées. S'il n'y a pas de correspondance dans la table de Flux :

- Le paquet est commuté vers le Contrôleur via le canal sécurisé OpenFlow.
- Le paquet sera Abandonné.
- Le paquet va continuer vers la table de Flux qui suit (Traitement Pipeline) Les instructions associées à chaque entrée décrivent l'acheminement du paquet, la modification, et les traitements dans le pipeline. Les instructions permettent lors d'un traitement par pipeline de
 - ✓ Envoyer le paquet vers la prochaine Table de Flux et la communication d'information entre elles sous forme de Meta-data.

- ✓ Le traitement s'arrête quand l'ensemble des instructions associées à une entrée ne spécifie plus de table, généralement arrivé à ce stade le paquet est modifié et acheminé vers la destination. [37]

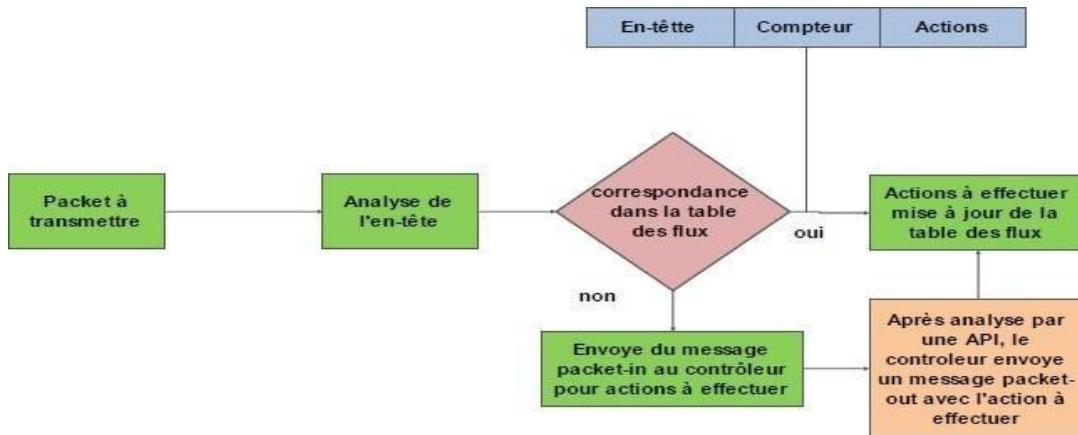


Figure 34- Processus de transmission des paquets au sein des commutateurs. [34]

2.2.12. Les Tables OpenFlow

Cette section décrit les composants de tables OpenFlow à savoir, les Table de Flux, les Tables de groupes, ainsi que les mécanismes de correspondance. [43]

2.2.12.1. Les Tables de Flux

Chaque Table de Flux dans un Switch contient un ensemble d'entrées.

Champ de Correspondance	Instructions	Compteurs
-------------------------	--------------	-----------

2.2.12.2. Les principaux éléments d'une entrée dans une table de Flux

- ✓ **Le champ de correspondance « Match Fields » :** c'est la partie sur laquelle le contrôleur se base pour faire correspondre les paquets, cela consiste à vérifier le port d'entrée ou l'en-tête du paquet afin d'y appliquer une action X.

- ✓ **Compteurs** : Il est possible de disposer d'un certain nombre de statistiques dont on sert pour la gestion des entrées dans les tables de flux.
- ✓ **Instructions** : c'est une opération pouvant contenir en elle-même un ensemble d'actions à appliquer sur le paquet.

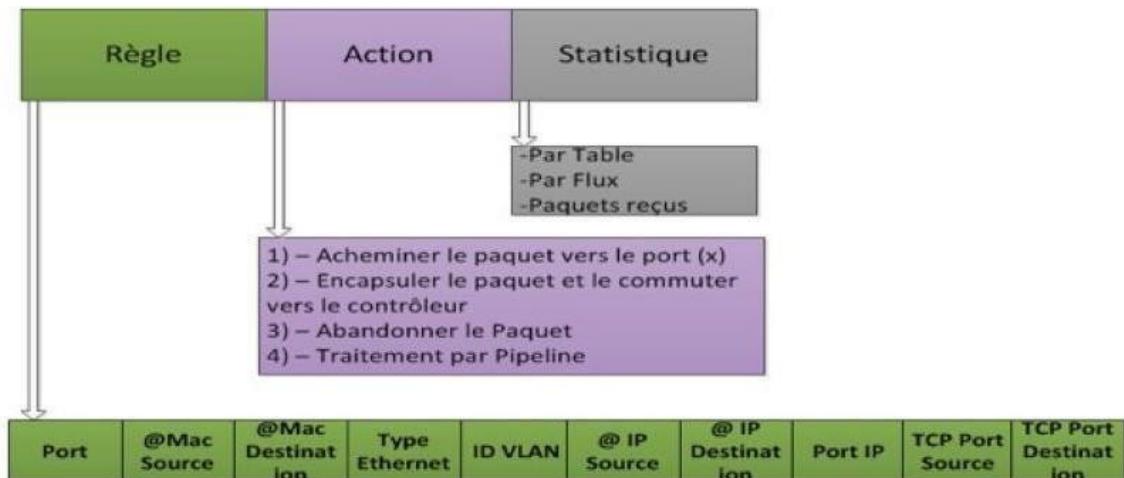


Figure 35- Contenu d'entrées de flux. [43]

2.2.13. Processus de traitement d'un paquet dans un Switch OpenFlow

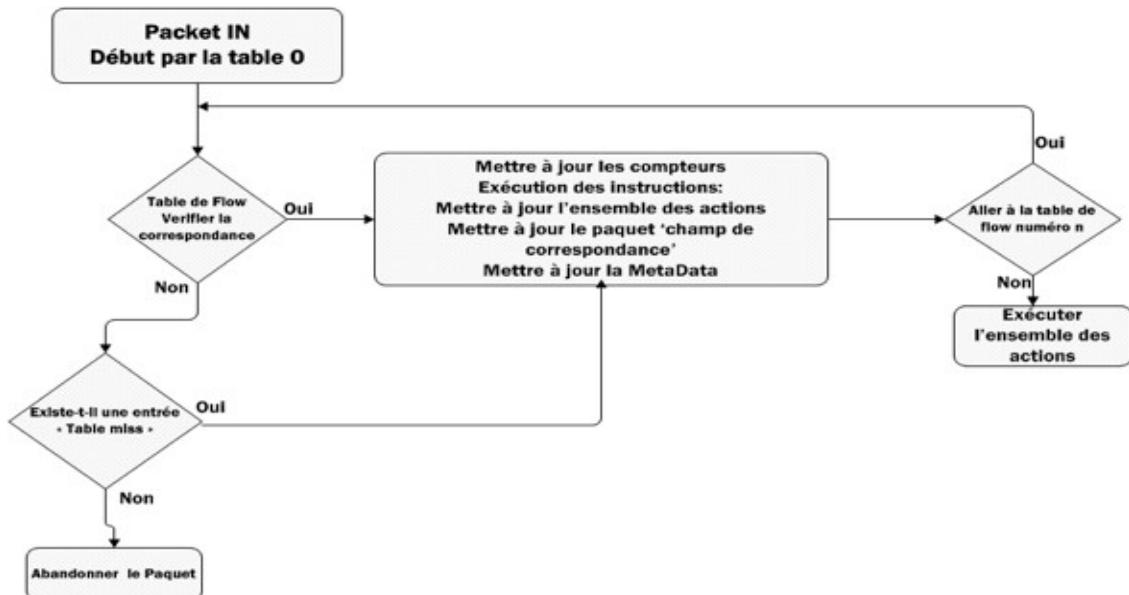


Figure 36- Processus de traitement d'un paquet [38].

A la réception d'un paquet, le Switch OpenFlow réalise les fonctions montrées dans les figures précédentes. Le Switch commence par faire une recherche dans la première table de Flux, et, en se basant sur le traitement par pipeline, il peut aussi effectuer une recherche dans les autres tables de Flux. Les instructions Chaque Entrée dans la table de flux contient un ensemble d'instructions qui seront exécutées quand un paquet correspond à cette entrée. Parmi les instructions supportées on trouve :

- **Apply-Action** : appliquer l'action spécifique immédiatement, sans aucun changement de l'Action set. Cette instruction peut être utilisée pour modifier le paquet entre deux tables de flux ou pour exécuter plusieurs actions du même type. Les actions sont spécifiées comme un action set
- **. Clear-Actions** : Effacer toutes les actions dans l'action set immédiatement.
- **Write-Actions** : Fusionner les actions dans l'action set actuel, si une des actions du même type existe déjà dans l'action set actuel, l'écraser, autrement l'ajouter.
- **Write Metadata** : Ecrire la valeur dans le champ Meta-data.
- **Goto-Table** : Indique la prochaine table de Flux dans le traitement pipe-line.

Action set

Un ensemble d'actions associées qui s'accumulent au fur et à mesure que le paquet avance dans la chaîne de traitement. Pipeline est traité par chaque table de flux. Quand une instruction ne contient pas un Goto-Table, le traitement pipeline s'arrête et les actions dans le set actions sont exécutées. Ce set est vide par défaut. Une entrée dans la table de Flux peut modifier l'action set en utilisant l'instruction Write-Action ou Clear-Action associée à un match. Quand plusieurs actions du même type sont requises, l'instruction Apply-Actions peut être utilisée. Les actions dans une action set sont appliquées dans l'ordre spécifié ci-dessous :

Traitement par Pipeline

- | | |
|--|--|
| <ul style="list-style-type: none">✓ Copy TTL inwards✓ Pop✓ Push✓ Copy TTL outwards✓ Decrement TTL | <ul style="list-style-type: none">✓ Set✓ Qos✓ Group✓ Output |
|--|--|

Pipeline OpenFlow : chaque Switch contient plusieurs tables de Flux, et chaque table contient plusieurs entrées. Le traitement pipeline OpenFlow décrit et définit comment les paquets interagissent avec les Tables de Flux

Les tables de Flux d'un Switch OpenFlow sont séquentiellement énumérées, commençant par 0. Le traitement par Pipeline commence toujours avec la première table de Flux : on fait correspondre le paquet selon les entrées de la table de Flux 0. D'autres tables de Flux peuvent être utilisées selon le résultat obtenu lors de la première correspondance faite dans la première table de Flux. Si le paquet correspond à une entrée dans la table de Flux, l'instruction en question sera exécutée. Les instructions dans les tables de Flux peuvent explicitement diriger le paquet vers une autre table de Flux utilisant l'instruction Goto. Une entrée x dans une Table de Flux n peut enchaîner le traitement du paquet en l'envoyant vers une autre table de Flux si seulement cette dernière dispose d'un numéro n supérieur à celui de la table où l'entrée x se trouve. Ainsi la dernière table du pipeline ne peut inclure l'instruction Goto. Si l'entrée dans la table de Flux ne redirige pas le paquet à une autre table de Flux, le traitement pipeline s'arrête. Arrivé à ce stade le paquet est traité avec l'action associée.[39]

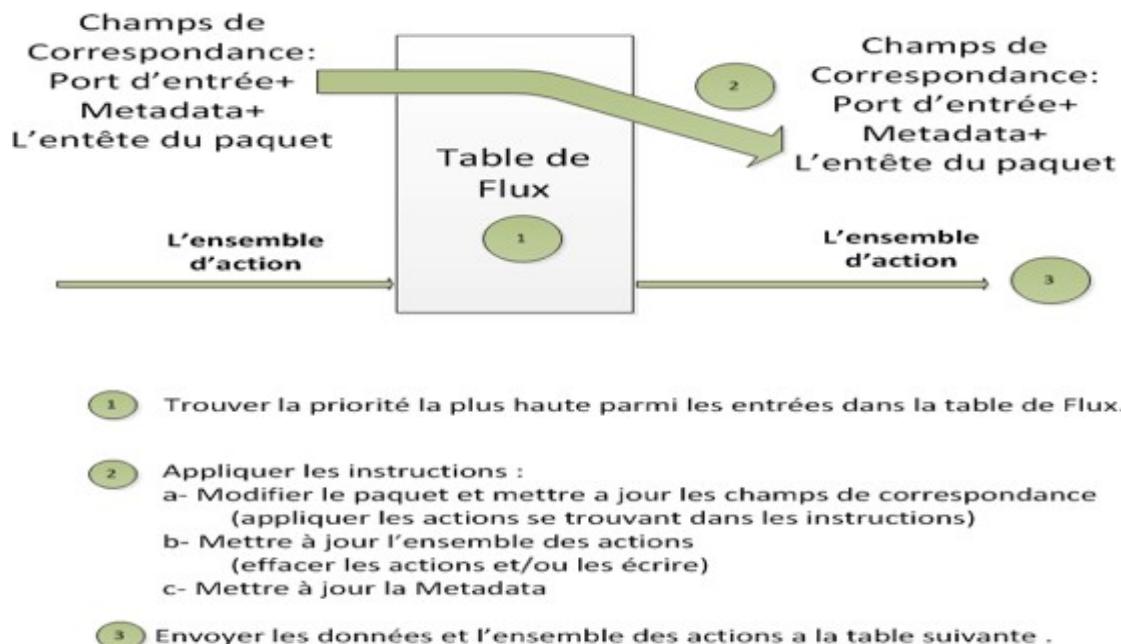


Figure 37- Traitement d'un paquet [39]

2.2.14. Les Tables de groupe

La table de groupe contient des entrées, et chaque entrée une liste d'actions appelée Conteneur D'actions. Les actions d'un ou plusieurs Conteneurs d'actions sont appliquées sur les paquets envoyés au groupe.

Chaque entrée dans la table de groupe contient :

Identifiant de Groupe	Type de Groupe	Compteurs	Conteneurs D'actions
-----------------------	----------------	-----------	----------------------

- **L'identifiant de groupe** : c'est un entier de 32 bits.
- **Le Type de groupe** : sert à déterminer le type du groupe « All – Select – Indirect – Fast Failover »
- **Les compteurs** : mis à jour quand un paquet est traité par un groupe.
- **Conteneur d'action** : un ensemble d'actions et de paramètres associés, qui sont définies pour les groupes.

2.2.15. Message OpenFlow

Le protocole OpenFlow supporte trois types de messages. Messages Contrôleur vers Switch, messages asynchrone et messages symétrique, chaque type a une sous-catégorie. [40]

2.2.15.1. Les Messages depuis le Contrôleur vers Switch

Sont initiés par le contrôleur, ils servent à gérer ou vérifier l'état du switch, ces types de messages peuvent ou non demander une réponse de la part du switch.

- **Features** : lors du Handshake le contrôleur peut demander l'identité et les capacités d'un switch en envoyant une requête.
- **Modify-State** : Ce type de message est envoyé pour gérer l'état dans les Switch. Sa fonction primaire est d'ajouter, modifier ou effacer les entrées dans les tables OpenFlow Flow/Groupe.
- **Read-state** : Ces messages sont utilisés par le contrôleur pour collecter différentes informations du switch, comme sa configuration actuelle, des statistiques et des capacités.

- **Packet-Out** : sont utilisés pour transférer les paquets reçus par les messages Packet-In. Soit ils contiennent le paquet en entier soit l'id du buffer faisant référence au paquet stocké dans le switch. Ils doivent contenir aussi une liste d'actions à appliquer, s'il n'y a pas d'action définie le paquet sera détruit.

a. Les Messages asynchrones

Les messages asynchrones sont envoyés par le switch vers le contrôleur pour indiquer un changement d'état ou l'arrivée d'un paquet.

- **Packet-In** : Avec ce type de message le switch transfère l'arrivée d'un paquet au contrôleur.
- **Flow-removed** : informe le contrôleur de la suppression d'une entrée dans la table de Flux
- **Port-Status** : Informe le contrôleur d'un changement sur un port du switch.

b. Les Messages symétriques

Les messages symétriques sont envoyés sans aucune sollicitation ni du switch ni du contrôleur.

- **Echo** : ont comme utilité la vérification de la connectivité entre switch et contrôleur.
- **Hello** : Ces messages sont échangés entre les deux une fois la connexion établie.
- **Error** : Utilisé pour signaler de part et d'autre des problèmes de connexion.

Test de connectivité

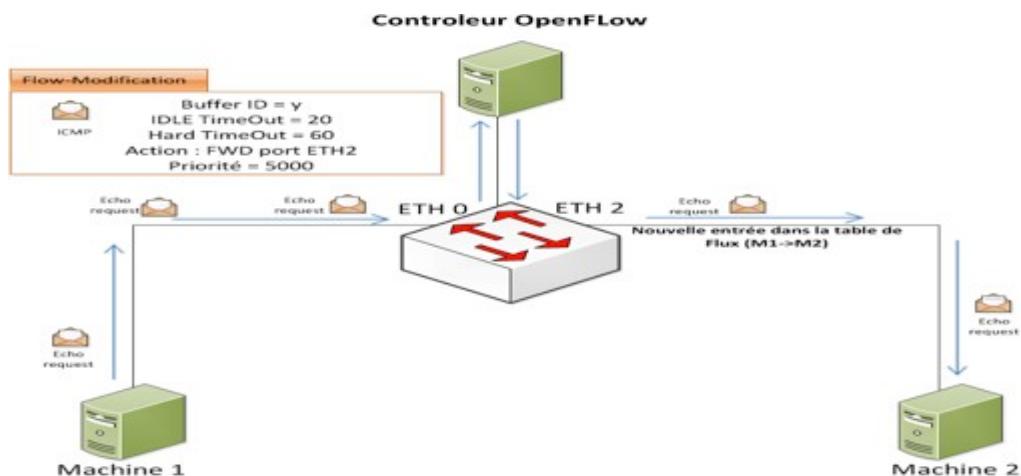


Figure 38- architecteur test de connectivité

2.2.16. Opportunités et défis des SDN

✓ Opportunités

L'approche SDN présente de multiples avantages pour faciliter et améliorer les tâches des administrateurs réseau. Nous allons détailler quelques-uns. [41]

✓ Réseau programmable

Avec SDN, il est plus simple de modifier les stratégies réseaux. La centralisation de la logique dans un contrôleur est entièrement personnalisée avec des connaissances globales et une puissance de calcul élevée, simplifient le développement des fonctions sophistiquées.

✓ La flexibilité

SDN apporte également une grande flexibilité dans la gestion du réseau. Il devient facile de rediriger le trafic, d'inspecter des flux particuliers, de tester de nouvelles stratégies ou de découvrir des flux intermédiaires.

✓ Le routage

SDN gère les informations de routage de manière centralisée en déléguant le routage et en utilisant une interface pour le contrôleur.

✓ Politique unifiée

SDN garanti une politique réseau unifiée et à jour grâce à son contrôleur car le contrôleur est responsable de l'ajout des règles dans les commutateurs.

✓ Gestion du cloud

SDN gère de manière très simple une plateforme cloud. La dynamique apportée par SDN traite des problèmes spécifiques au cloud tels que l'évolutivité, l'adaptation, ou des mouvements de machines virtuels.

✓ Simplification matérielle

SDN a tendance à utiliser des technologies standards et de bases pour contrôler les équipements du réseau, tandis que la puissance de calcul n'est requise qu'au niveau du contrôleur. Ainsi, les équipements de réseau deviendront des produits à bas prix offrant des

interfaces standard. Avec ce type de matériel, il serait également simple d'ajouter de nouveaux périphériques, puisqu'ils ne sont pas spécialisés, de les connecter au réseau et de laisser le contrôleur les gérer conformément à la politique définie. Ainsi le réseau devient facilement évolutif dès que le contrôleur est évolutif.

2.2.17. Les défis des réseaux SDN

Les réseaux SDN ont connu plusieurs défis que ce soit sur le plan de données ou sur le plan de contrôle. Pour notre travail, nous allons présenter des défis au niveau du plan de contrôle. Ces derniers comprennent la performance, la scalabilité, la fiabilité et la sécurité. [41]

2.2.17.1. La performance

La performance des contrôleurs SDN reste un domaine très important. Depuis l'arrivée des SDN, les chercheurs essayent toujours d'améliorer les performances des contrôleurs. La technique de SDN est basée sur les flux, du coup ces performances sont mesurées en fonction de deux métriques : le temps nécessaire pour instaurer un nouveau flux dans les commutateurs et le nombre de flux que le contrôleur peut traiter par seconde.

2.2.17.2. La scalabilité

La scalabilité ou l'évolutivité du réseau SDN est un autre défi que présente ce paradigme. Comme nous le savons, plus la taille du réseau augmente, plus des demandes sont envoyées au contrôleur. Donc à un moment donné, le contrôleur peut être incapable de traiter toutes les demandes.

2.2.17.3. La fiabilité

On fait allusion aux premiers déploiements des SDN. Un seul contrôleur est utilisé, et ce dernier est responsable de tout le réseau. S'il tombe en panne ou devient défaillant alors c'est tout le réseau qui devient indisponible.

2.2.17.4. La sécurité

Les SDN présentent un défi majeur pour la sécurité. Le fait que toute l'intelligence du réseau soit centralisée en un point qui est le contrôleur peut rendre ce dernier vulnérable. Si le contrôleur devient indisponible ou compromis tous les aspects du réseau seront endommagés c'est pourquoi il représente un point critique. Le réseau SDN est soumis à plusieurs problèmes de sécurité tels que le déni de service, l'usurpation d'identité, l'élévation des priviléges, la falsification, la répudiation, etc.

2.2.18. Impacte des réseaux SDN

Malgré tous ses avantages, l'approche SDN a des impacts importants dans l'utilisation des réseaux. On ne peut dire des inconvénients mais des changements qui nécessitent un coût. Nous allons en citer quelques un : Un problème de dotation qui veut dire une reformation du personnel ou recrutement de nouvel agent pour bien se familiariser à son utilisation. Il faut une très grande réorganisation, des partages, d'information entre entreprises, d'applications, serveurs et équipes réseautages. Comme nous l'avons évoqué ci-dessus, intervient le problème de coût pour le recyclage, une réorganisation, de nouvelle licence, perte de continuité des activités lors du déploiement initial. Et pour la sécurité, on peut noter que c'est une nouvelle technologie, avec de nouveau protocoles qui peut aboutir à des faiblesses et de vulnérabilité. Et pour finir Le SDN ne permet pas de : réduire la dépendance vis-à-vis de la topologie spécifique du matériel physique sous-jacent, virtualiser toutes les fonctions et tous les composants réseau, déployer des réseaux en parallèle avec des ressources de calcul et de stockage virtualisés. [43]

2.3. Conclusion

La solution SDN offre des avantages significatifs pour les réseaux programmables, en permettant une gestion plus efficace et sécurisé des réseaux d'objets connectées, une bonne connectivité et meilleure gestion de la bande passante. Dans ce chapitre on a introduit le concept de SDN en expliquant les différentes couches de l'architecture de SDN. Le but était de montrer que cette approche permettait de rendre les réseaux programmables, évolutifs et faciles à jouer. Dans le chapitre suivant, nous présentons les différents composants d'un réseau SDN

Chapitre 3

LA MISE EN PLACE ET RESULTATS

3.1. Introduction

Dans ce chapitre nous allons apporter notre solution en tenant compte de l'existant, en améliorant l'architecture existante et en ramenant de la nouveauté du point de vue sécuritaire fiabilité et performance.

3.2. Architecture proposée

C'est une topologie SDN dans un réseau local composée d'un contrôleur SDN opendaylight avec 6 commutateurs openflow

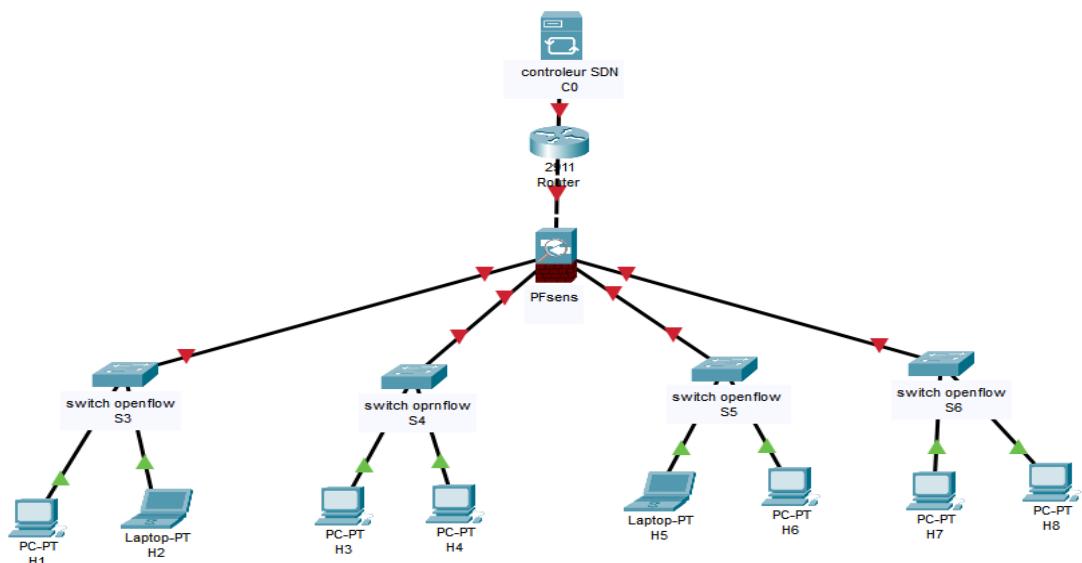
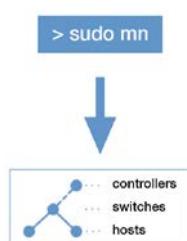


Figure 39- la topologie SDN mise en place

Pour cette partie nous avons choisi de faire l'étude avec le contrôleur opendaylight qui nous à donner de bons résultats dans l'étude comparative

3.3. Outils & Technologies

❖ Minnet



Mininet est un émulateur de réseau qui crée un réseau d'hôtes virtuels, de commutateurs, de contrôleurs et de liens. Les hôtes Mininet exécutent un logiciel réseau Linux standard et ses commutateurs prennent en charge OpenFlow pour un routage personnalisé très flexible et une mise en réseau définie par logiciel.

❖ dlux



Le module utiliser par opendaylight pour l'affichage graphique de topology , flowtable et plusieurs features.

❖ vmware workstation



VMware Workstation est une gamme de produits Desktop Hypervisor qui permettent aux utilisateurs d'exécuter des machines virtuelles, des conteneurs et des clusters Kubernetes. En savoir plus sur le produit.

❖ **solarputty**



Est un émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet, rlogin, et TCP brut. Il permet également des connexions directes par liaison série RS-232. À l'origine disponible uniquement pour Windows, il est à présent porté sur diverses plates-formes Unix

❖ **postman**



Postman facilite la création et l'envoi de requêtes API. Envoyez une demande pour tester un endpoint, récupérer des données à partir d'une source de données ou essayer la fonctionnalité d'une API.

3.4. Taches

1. Configurer la machine virtuelle Mininet .
2. Installez OpenDaylight sur Ubuntu 20.04
3. création de la topologie SDN.
4. envoyer des requêtes openflow à partir de la couche d'application

3.4.1. Présentation de l'environnement de travail

3.4.1.1. Installation de VMware Workstation pro

VMware Workstation est un outil de virtualisation il permet de créer de nouvelles machines virtuelles, transformer un PC en une machine virtuelle et effectuer un déploiement en masse.Afin de créer les machines utilisateurs virtuelles au sein du même pc, nous sommes appelés à installer VMware Workstation en suivant les étapes d'installations jusqu'à la fin puis cliquer sur le bouton « **terminer** »

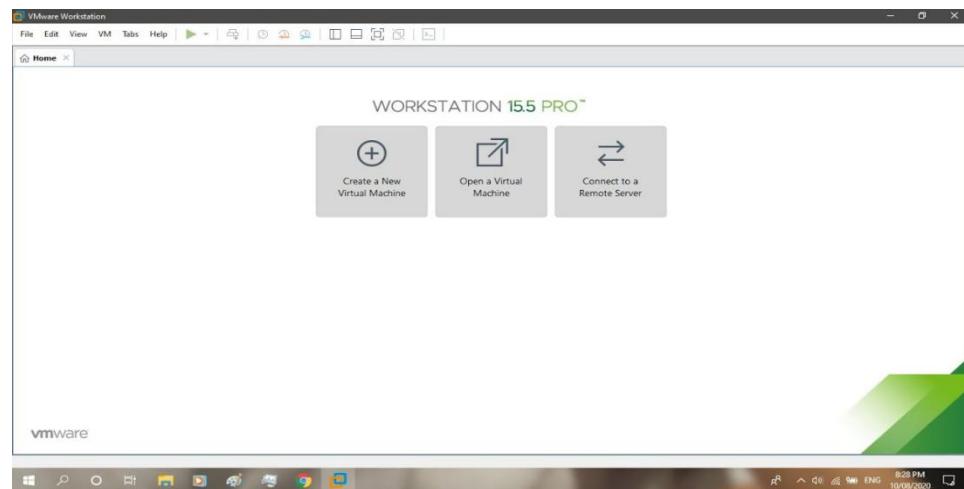
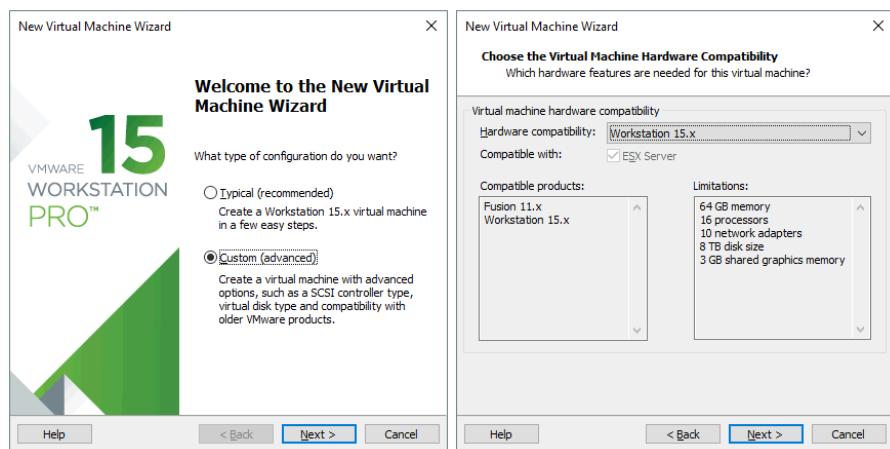


Figure 40 : L'interface graphique de VMware Workstation pro.

3.4.1.2. Configurer la machine virtuelle Mininet

➤ Télécharger mininet VM

Option 1 : Installation de la VM Mininet (facile, recommandée)

L'installation d'une VM est le moyen **le plus simple et le plus infaillible** d'installer Mininet, c'est donc ce par quoi nous vous recommandons de commencer.

Suivez ces étapes pour une installation de VM :

1. Téléchargez une **image de machine virtuelle Mininet** à partir [des versions Mininet](#).
2. Téléchargez et installez un **système de virtualisation**. Nous recommandons l'une des options gratuites suivantes :
 - [VirtualBox](#) (GPL, macOS/Windows/Linux)
 - [VMware Fusion](#) (macOS)
 - [Lecteur VMware Workstation](#) (Windows/Linux)

Vous pouvez également utiliser l'un des éléments suivants :

- [Qemu](#) (gratuit, GPL) pour n'importe quelle plateforme
- Microsoft [Hyper-V](#) (Windows)
- [KVM](#) (gratuit, GPL, Linux)

Mininet 2.3.0 Dernier

Mininet 2.3.0

Annonce de sortie :
<http://mininet.org/blog/2021/02/28/announcing-mininet-2-3-0/>

Notes de version :
<https://github.com/mininet/mininet/wiki/Mininet-2.3.0-Release-Notes>

Instructions de téléchargement et de mise à niveau :
<http://mininet.org/download/>

Sommes de contrôle des images de VM :
[sha256sum-2.3.0-bionic64server.txt](#)
[sha256sum-2.3.0-focal64server.txt](#)
[sha256sum-2.3.0-xenial64server.txt](#)
[sha256sum-2.3.0-xenial32server.txt](#)

Images de machine virtuelle :
Des images de VM prédéfinies sont disponibles pour les versions récentes d'Ubuntu LTS.
Nous recommandons l' [image de la machine virtuelle Ubuntu 20.04.1](#), sauf si vous avez besoin d'une version antérieure.
Voir « Actifs » ci-dessous

Figure 41 : Téléchargement Mininet

➤ **Créer la machine virtuelle Mininet**

sur VMware workstation taper sur open:

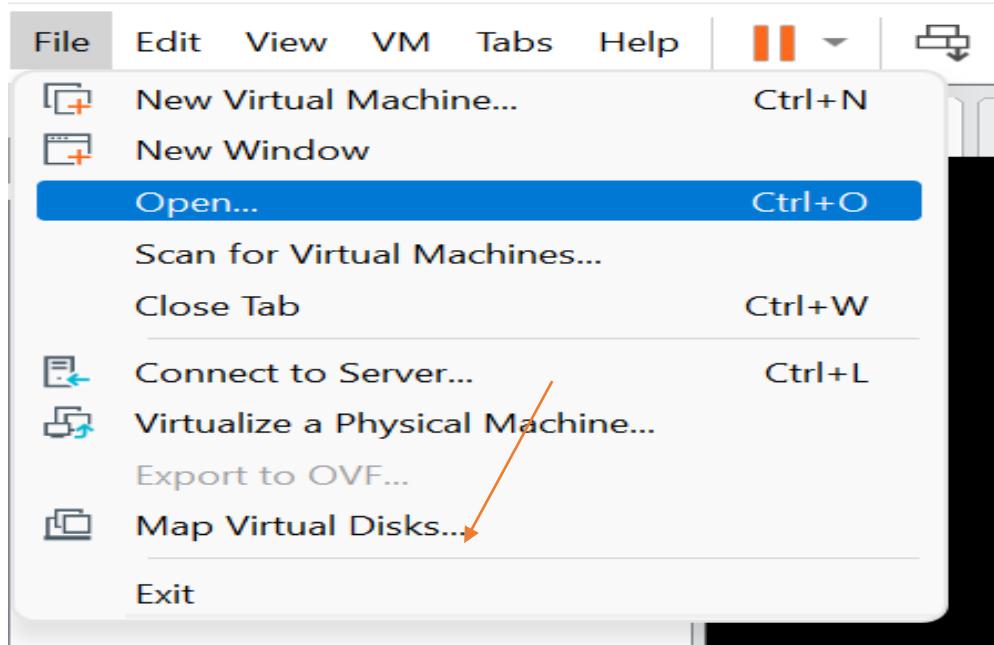


Figure 42: Crée la machine virtuelle Mininet

➤ **Choisir Mininet VM**

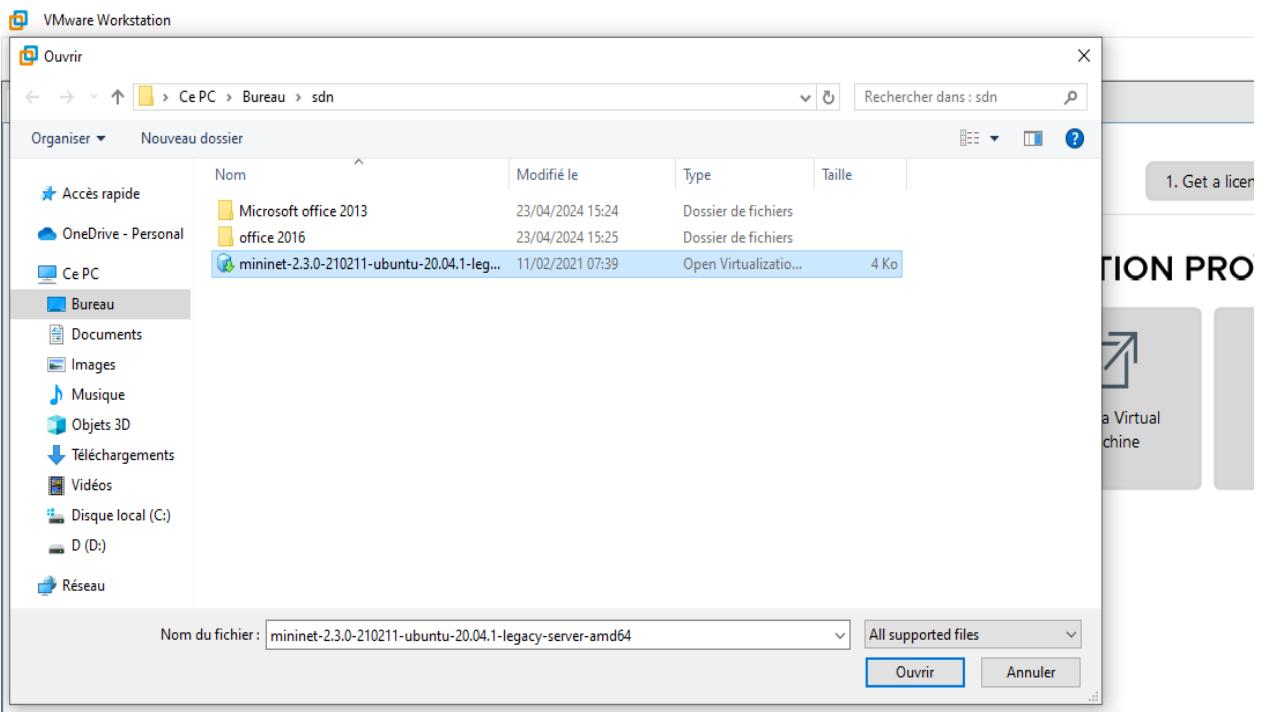


Figure 43: ouvrir Mininet

➤ La configuration nécessaire

Une fois VMware lancé, nous devons d'abord augmenter l'espace du disque de la VMware comme l'indique la Figure



Figure 44: configuration Mininet

➤ Démarrer mininet

Login : mininet

password : mininet

```
Ubuntu 20.04.1 LTS mininet-vm tty1

mininet-vm login: mininet
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

Figure 45 :Démarrer Mininet

➤ Tester mininet

pour activer mininet taper la commande : **sudo mn**

```
mininet@mininet-vm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> _
```

Figure 46 :Tester Mininet

➤ Trouver adresse Mininet

```
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.7.241 netmask 255.255.255.0 broadcast 192.168.7.255
              ether 00:0c:29:cb:63:8e txqueuelen 1000 (Ethernet)
                    RX packets 52 bytes 5483 (5.4 KB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 83 bytes 7521 (7.5 KB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              loop txqueuelen 1000 (Local Loopback)
                    RX packets 5 bytes 390 (390.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 5 bytes 390 (390.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$
```

Figure 47:adresse Mininet.

3.4.1.3. Installez opendaylight sur ubuntu 20.04

Tout d'abord je choisis le serveur ubuntu pour y installer opendaylight

➤ Télécharger l'image iso d'ubuntu server

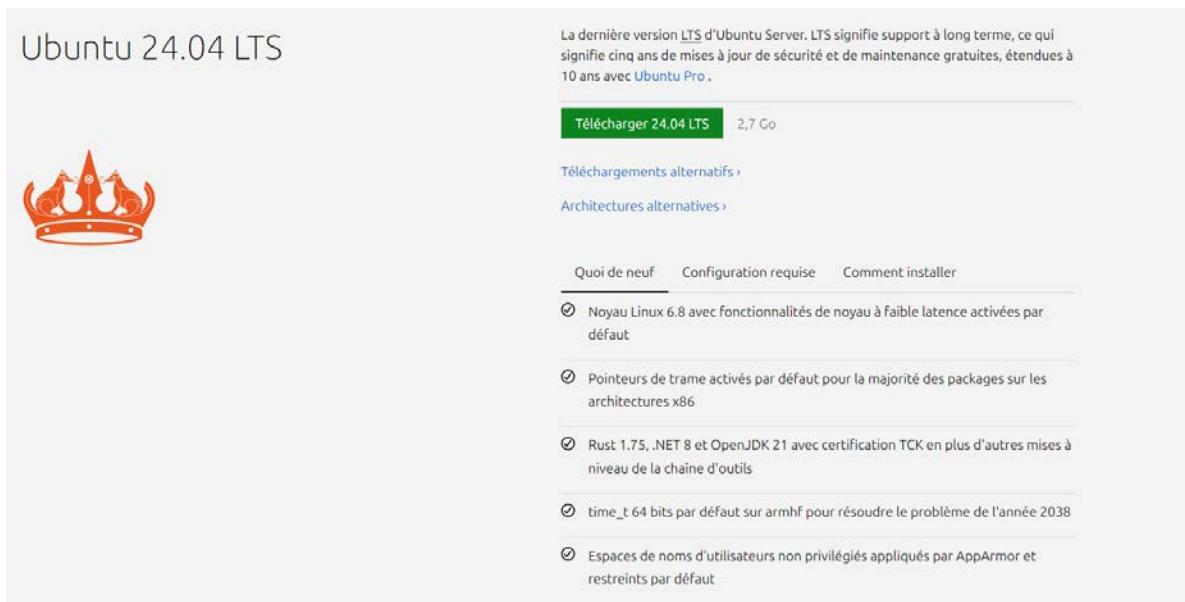


Figure 48:télécharger l'image iso d'ubuntu server

➤ Créer une machine virtuelle avec cette image iso

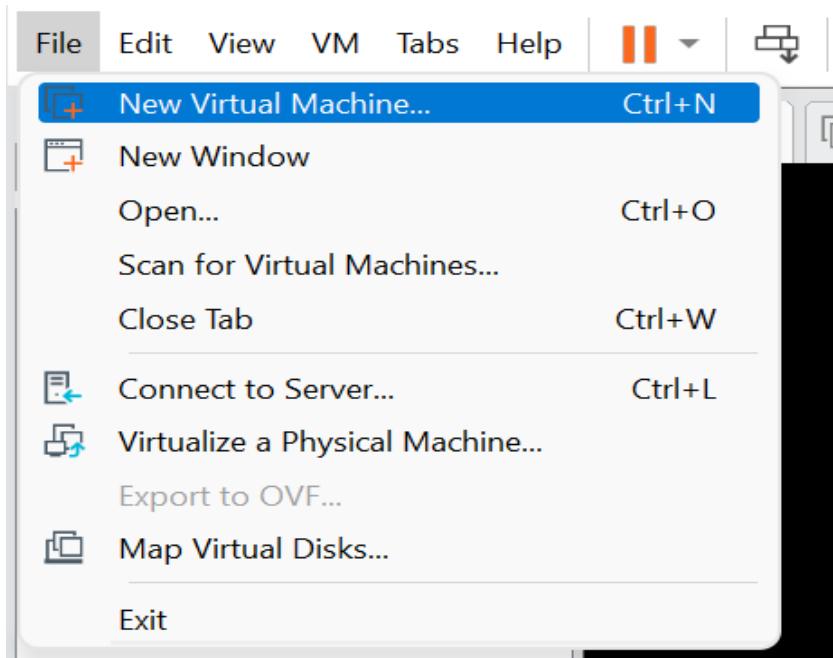


Figure 49: création une machine virtuel ODL

➤ Configuration nécessaire

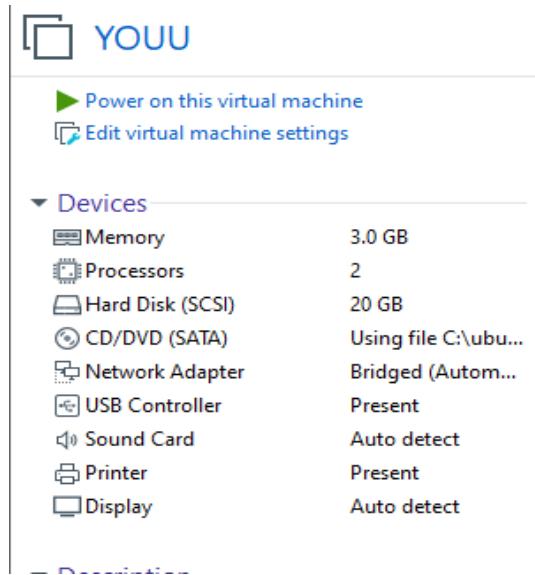


Figure 50 : configuration ODL

➤ Installer openDayLight sur ubuntu server VM

- Utilisé un émulateur de terminal

L'utilisation direct sur la machine virtuelle ubuntu server n'est pas de tout pratique par exemple on peut pas utiliser copie coller et on peut pas modifier.

Pour trouver adresse ip de cette machine taper **ifconfig**

```
yubasdn@serverod:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.7.106 netmask 255.255.255.0 broadcast 192.168.7.255
        inet6 fe80::20c:29ff:fe48:7578 prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:48:75:78 txqueuelen 1000 (Ethernet)
            RX packets 76349 bytes 42410146 (42.4 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 82159 bytes 8424268 (8.4 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 464 bytes 42427 (42.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 464 bytes 42427 (42.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

yubasdn@serverod:~$ _
```

Figure 51 : affichage adresse ip Mininet

➤ Sur solar putty

- #### • **Créer une nouvelle session**

Assurez-vous que le nom et mot de passe sont les même que la machine virtuel 24

EDIT SESSION DETAILS

Session name

IP or hostname

 Port

Type of connection

 ▼

Use credentials

 ▼

CUSTOMIZATION

Set custom color of this session

Use post-authenticate script

Enable session logging

Save Cancel

Figure 52 : session ODL

- Taper la commande ifconfig pour rassurer l'adresse de contrôleur

```
yubasdn@serverod:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.7.106 netmask 255.255.255.0 broadcast 192.168.7.255
      inet6 fe80::20c:29ff:fe48:7578 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:48:75:78 txqueuelen 1000 (Ethernet)
          RX packets 76418 bytes 42416544 (42.4 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 82194 bytes 8429263 (8.4 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 494 bytes 44820 (44.8 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 494 bytes 44820 (44.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

yubasdn@serverod:~$
```

Figure 53 : adresse IP ODL

- Préparer le système d'exploitation

Sur Cette session exécutez : une mise à jour **apt-get**, qui actualise la liste des packages disponibles.

```
yubasdn@serverod:~$ 
yubasdn@serverod:~$ sudo apt-get update
```

- Installez unzip :

Pour décompresser l'archive OpenDaylight :

```
yubasdn@serverod:~$ sudo apt-get -y install unzip vim wget
```

- Installe JAVA jre

Les architectes OpenDaylight ont conçu OpenDaylight pour l'écosystème Java. OpenDaylight nécessite un environnement Java (JRE) pour fonctionner.

➤ Pour que l'ODL fonctionne il faut télécharger la version 8 de java

```
yubasdn@serverod:~$ sudo apt-get -y install openjdk-8-jre
yubasdn@serverod:~$ sudo update-alternatives --config java
yubasdn@serverod:~$ sudo echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre' >> ~/.bashrc
yubasdn@serverod:~$ source ~/.bashrc
```

➤ Pour vérifier si le JAVA_HOME est correctement configuré

Vous pouvez suivre les étapes suivantes : Ouvrir un terminal (ou une invite de commande sur Windows). Saisir la commande suivante pour afficher la valeur de JAVA_HOME :

```
yubasdn@serverod:~$ echo $JAVA_HOME
```

➤ Télécharger le fichier d'opendaylight

```
yubasdn@serverod:~$ wget https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integrati
on/karaf/0.8.4/karaf-0.8.4.zip
```

➤ Créer un nouveau dossier sur usr/local pour décompresser le fichier karaf

```
yubasdn@serverod:~$ sudo mkdir /usr/local/karaf
```

➤ Déplacer OpenDaylight téléchargé vers le répertoire Karaf

```
yubasdn@serverod:~$ sudo mv karaf-0.8.4.zip /usr/local/karaf  
yubasdn@serverod:~$ sudo unzip /usr/local/karaf/karaf-0.8.4.zip -d /usr/local/karaf/  
yubasdn@serverod:~$ sudo update-alternatives --install /usr/bin/karaf karaf /usr/local/karaf/karaf-0.8.4/bin/karaf 1
```

➤ Install et configuration des chemins pour le fichier karaf

```
yubasdn@serverod:~$ sudo update-alternatives --config karaf  
yubasdn@serverod:~$ which karaf
```

➤ Executer OpenDaylight avec la commande suivante

```
yubasdn@serverod:~$ sudo -E karaf
```

➤ Démarrer opendaylight

```
link: /etc/alternatives/karaf  
link: /usr/local/karaf/karaf-0.8.4/bin/karaf  
Apache Karaf starting up. Press Enter to open the shell now...  
100% [=====]  
Karaf started in 62s. Bundle stats: 460 active, 461 total  
  
Hit '<tab>' for a list of available commands  
and '[cmd] --help' for help on a specific command.  
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.  
opendaylight-user@root>
```

Figure 54 Démarrer opendaylight

➤ Installer features besoin pour une installation excellente de opendaylight

```
opendaylight-user@root>feature:install odl-l2switch-switch-ui
opendaylight-user@root>feature:install odl-restconf
opendaylight-user@root>feature:install odl-mdsal-apidocs
opendaylight-user@root>feature:install odl-dlux-core
opendaylight-user@root>feature:install odl-dluxapps-topology
opendaylight-user@root>feature:install odl-dluxapps-nodes
```

Une fois ces fonctionnalités installées avec succès. Accédez à l'interface utilisateur dlux à l'adresse : http://<controller_ip@>:8181/index.html.

➤ Dans notre cas notre adresse dlux est <http://192.168.7.106:8181/index.html>.

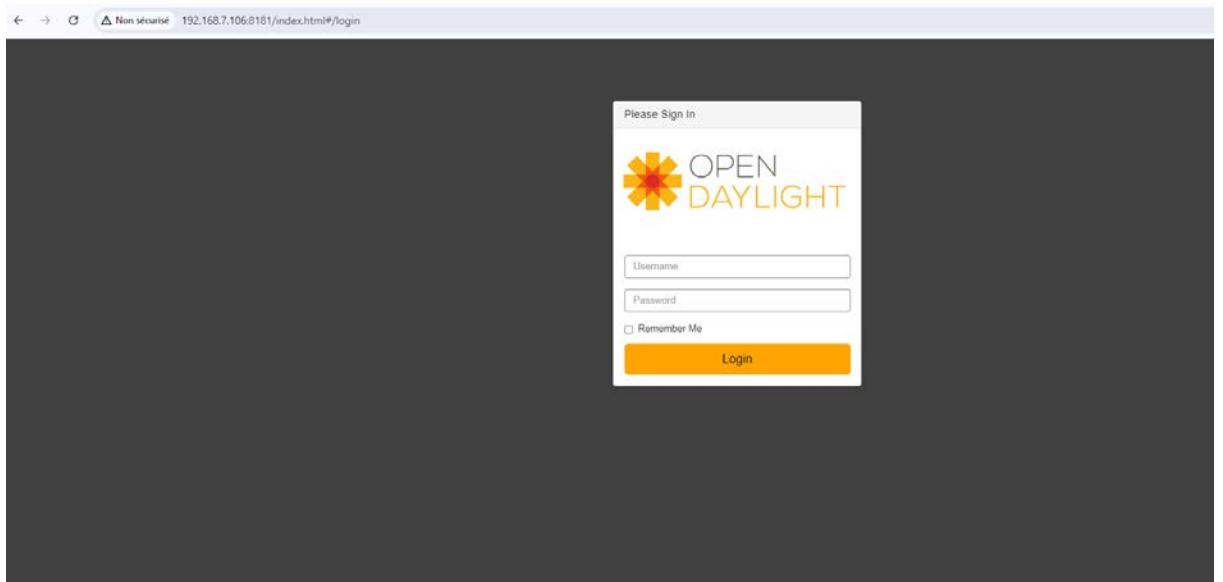


Figure 55 : Accès à l' interface utilisateur dlux opendaylight

➤ **Pour accéder à l'intérieur de l'interface dlux opendaylight**

On utilise les informations d'identification par défaut sont admin/admin pour la connexion.

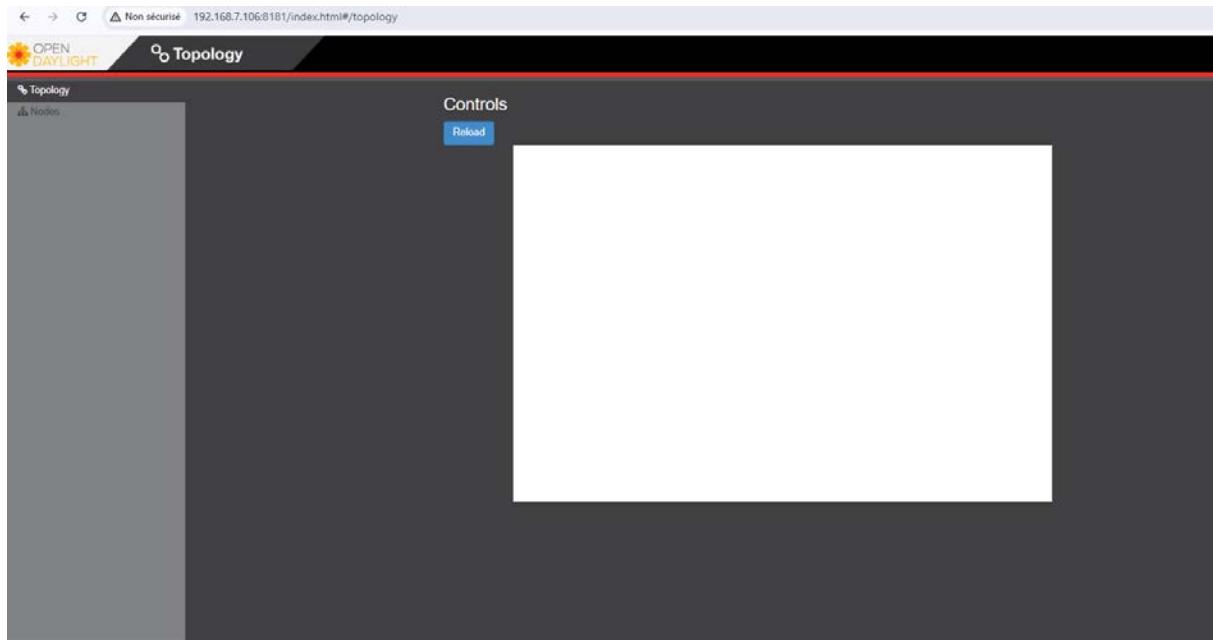


Figure 56 : l'interface dlux opendaylight

3.4.1.4. Crédation de la topologie SDN

Après télécharger mininet (partie infrastructure) et openDayLight Controller (partie contrôleur) on va créer notre topologie SDN :

1- test la connectivité entre mininet et openDayLight Controller

L'adresse ip de mininet VM est : 192.168.144.128

L'adresse ip de contrôleur ODL est : 192.168.144.134

Mininet

```
yubasdn@serverod:~$ ping 192.168.7.106
PING 192.168.7.106 (192.168.7.106) 56(84) bytes of data.
64 bytes from 192.168.7.106: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 192.168.7.106: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 192.168.7.106: icmp_seq=3 ttl=64 time=0.029 ms
64 bytes from 192.168.7.106: icmp_seq=4 ttl=64 time=0.036 ms
64 bytes from 192.168.7.106: icmp_seq=5 ttl=64 time=0.028 ms
^C
--- 192.168.7.106 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4112ms
rtt min/avg/max/mdev = 0.020/0.045/0.113/0.034 ms
yubasdn@serverod:~$ c
```

Figure 57 : L'adresse ip de mininet VM

ODL

```
yubasdn@serverod:~$ ping 192.168.7.241
PING 192.168.7.241 (192.168.7.241) 56(84) bytes of data.
64 bytes from 192.168.7.241: icmp_seq=1 ttl=64 time=0.636 ms
64 bytes from 192.168.7.241: icmp_seq=2 ttl=64 time=0.620 ms
64 bytes from 192.168.7.241: icmp_seq=3 ttl=64 time=4.79 ms
64 bytes from 192.168.7.241: icmp_seq=4 ttl=64 time=0.383 ms
64 bytes from 192.168.7.241: icmp_seq=5 ttl=64 time=1.10 ms
^C
--- 192.168.7.241 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4136ms
rtt min/avg/max/mdev = 0.383/1.505/4.787/1.657 ms
yubasdn@serverod:~$ 
```

Figure 58 :L'adresse ip de ODL VM

➤ Crée la topologie sur mininet

```
GNU nano 4.8                                     mytopo.py
from mininet.net import Mininet
from mininet.node import RemoteController, OVSSwitch
from mininet.link import TCLink
from mininet.cli import CLI
from mininet.log import setLogLevel

def myTopology():

    # Create an empty network and add nodes to it

    net = Mininet(controller=RemoteController, switch=OVSSwitch, link=TCLink)

    # Add the remote controller

    c0 = net.addController('c0', controller=RemoteController, ip='192.168.7.106', port=6653)

    # Add switch

    s1 = net.addSwitch('s1') # Router
    s2 = net.addSwitch('s2') # PfSense
    s3 = net.addSwitch('s3') # DLink 24 (1)
    s4 = net.addSwitch('s4') # DLink 24 (2)
    s5 = net.addSwitch('s5') # DLink 24 (3)
    s6 = net.addSwitch('s6') # DLink 24 (4)

    # Add hosts

    h1 = net.addHost('h1', ip='10.0.0.1')
    h2 = net.addHost('h2', ip='10.0.0.2')
    h3 = net.addHost('h3', ip='10.0.0.3')
    h4 = net.addHost('h4', ip='10.0.0.4')
    h5 = net.addHost('h5', ip='10.0.0.5')
    h6 = net.addHost('h6', ip='10.0.0.6')

    # Get Help   ^W Write Out   ^T Where Is   ^X Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
    ^X Exit      ^R Read File   ^R Replace   ^U Paste Text  ^T To Spell   ^G Go To Line M-E Redo
```

```
GNU nano 4.8                                     mytopo.py                                     Modified
# Add hosts

h1 = net.addHost('h1', ip='10.0.0.1')
h2 = net.addHost('h2', ip='10.0.0.2')
h3 = net.addHost('h3', ip='10.0.0.3')
h4 = net.addHost('h4', ip='10.0.0.4')
h5 = net.addHost('h5', ip='10.0.0.5')
h6 = net.addHost('h6', ip='10.0.0.6')
h7 = net.addHost('h7', ip='10.0.0.7')
h8 = net.addHost('h8', ip='10.0.0.8')

# Add Links between switches

net.addLink(s1, s2)
net.addLink(s2, s3)
net.addLink(s2, s4)
net.addLink(s2, s5)
net.addLink(s2, s6)

# connect hosts to switches

net.addLink(h1, s3)
net.addLink(h2, s3)
net.addLink(h3, s4)
net.addLink(h4, s4)
net.addLink(h5, s5)
net.addLink(h6, s5)
net.addLink(h7, s6)
net.addLink(h8, s6)

# start the network

# Get Help   ^W Write Out   ^T Where Is   ^X Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File   ^R Replace   ^U Paste Text  ^T To Spell   ^G Go To Line M-E Redo
```

The screenshot shows a terminal window with the nano text editor open. The file is named 'mytopo.py'. The code in the file is as follows:

```
GNU nano 4.8           mytopo.py           Modified

# Add Links between switches
net.addLink(s1, s2)
net.addLink(s2, s3)
net.addLink(s2, s4)
net.addLink(s2, s5)
net.addLink(s2, s6)

# connect hosts to switches
net.addLink(h1, s3)
net.addLink(h2, s3)
net.addLink(h3, s4)
net.addLink(h4, s4)
net.addLink(h5, s5)
net.addLink(h6, s5)
net.addLink(h7, s6)
net.addLink(h8, s6)

# start the network
net.start()

# Wait for the controller to be fully ready
CLI(net)
net.stop()
if __name__ == '__main__':
    setLogLevel('info')
    myTopology()

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^U Undo
^X Exit      ^R Read File  ^P Replace   ^U Paste Text ^T To Spell  ^L Go To Line ^E Redo
```

Figure 59 : script python

Taper la commande **ctrl+O** pour enregistrer le script python (mytopo.py) ensuite **ctrl+x** pour revenir au terminal mininet.

➤ Executer le script python

Taper la commande **sudo python3 mytopo.py** sur mininet vm

The screenshot shows a terminal window on a 'mininet@mininet-vm' session. The command **sudo python3 mytopo.py** is run, and the output is as follows:

```
mininet@mininet-vm:~$ sudo python3 mytopo.py
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
*** Starting CLI:
mininet> _
```

Figure 60 : Exécution du script python

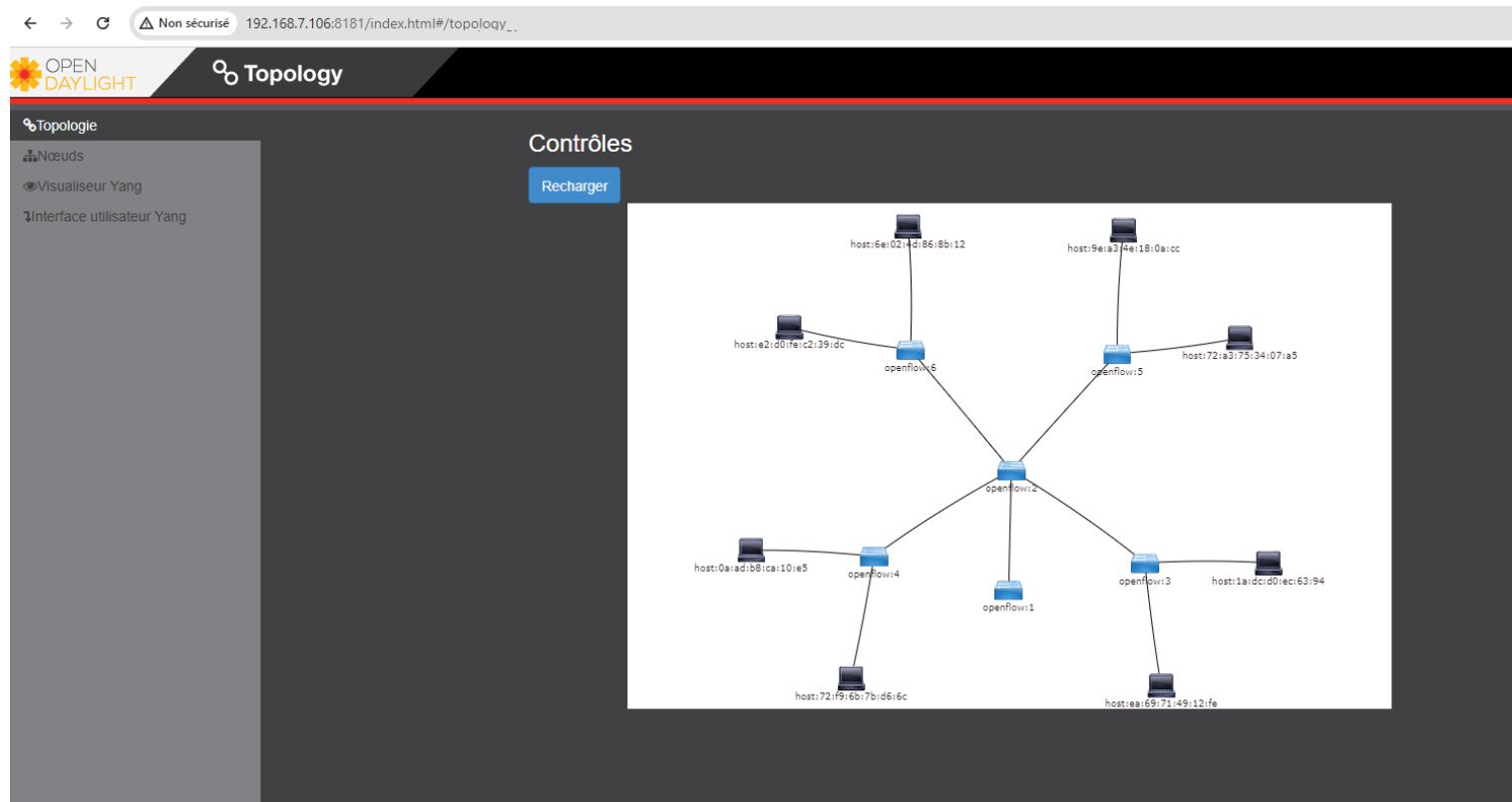
➤ Tester connectivité

Taper la commande **pingall**

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Results: 0% dropped (56/56 received)
mininet> _
```

Figure 61 : tester la connectivité

➤ Voir la topologie sur GUI openDaylight



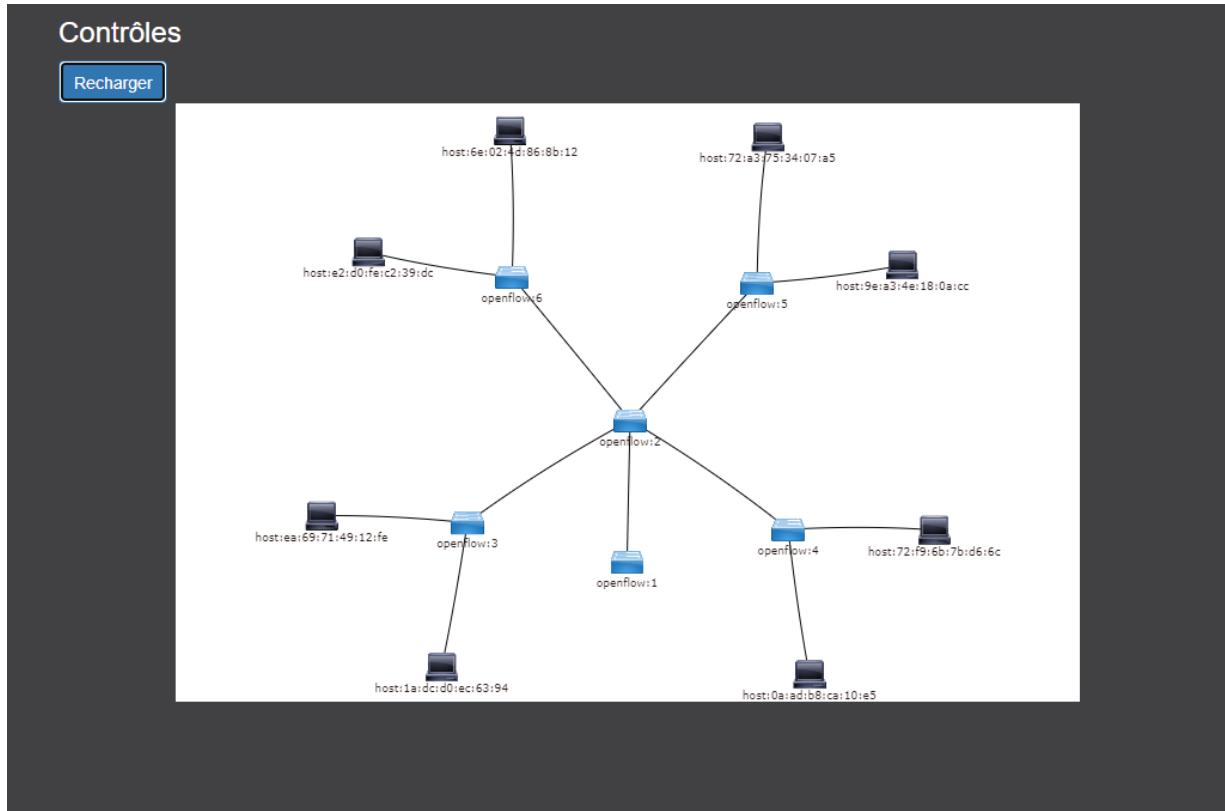


Figure 62 : topologie.opendaylight

- La configuration de base des différents hôtes est donnée par la table suivante :

Hotes	Adresse MAC	Adresse IP
H1	ea :69 :71 :49 :12 :fe	10.0.0.0.1
H2	1a :dc :d0 :ec :63 :94	10.0.0.0.2
H3	0a:ad:b8:ca:10:e5	10.0.0.0.3
H4	72 :f9 :6b :7b :d6 :C6	10.0.0.0.4
H5	9e :a3 :4e :18 :0a :cc	10.0.0.0.5
H6	72 :a3 :75 :34 :07 :a5	10.0.0.0.6
H7	6e :02 :4d :86 :8b :12	10.0.0.0.7
H8	e2 :d0 :fe :c2 :39 :dc	10.0.0.0.8

Tableau 4 : Informations Hôtes

➤ Tester la connectivité entre les hots

Sur terminal mininet command :

h1 ping h2

```
mininet> h1 ping 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=0.235 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.138 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4076ms
rtt min/avg/max/mdev = 0.052/0.121/0.235/0.063 ms
mininet>
```

Figure 63 : ping hote 1 vers hote 2

h1 ping -c3 h2

```
mininet> h1 ping -c3 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=0.047 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.044 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.044/0.045/0.047/0.001 ms
mininet>
```

Figure 64 : ping hote 1 vers hote 2 seulement pour les 3 premier paquet

➤ **Lister les nodes**

Node Id	Node Name	Node Connectors	Statistics
openflow:4	s4	4	Flows Node Connectors
openflow:5	s5	4	Flows Node Connectors
openflow:6	s6	4	Flows Node Connectors
openflow:1	s1	2	Flows Node Connectors
openflow:2	s2	6	Flows Node Connectors
openflow:3	s3	4	Flows Node Connectors

Figure 65: Liste des nodes

➤ **Nodes sur mininet**

```
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 h7 h8 s1 s2 s3 s4 s5 s6
mininet>
```

Figure 66 : Affichage des nodes sur mininet

➤ **dump**

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=4088>
<Host h2: h2-eth0:10.0.0.2 pid=4090>
<Host h3: h3-eth0:10.0.0.3 pid=4092>
<Host h4: h4-eth0:10.0.0.4 pid=4094>
<Host h5: h5-eth0:10.0.0.5 pid=4096>
<Host h6: h6-eth0:10.0.0.6 pid=4098>
<Host h7: h7-eth0:10.0.0.7 pid=4100>
<Host h8: h8-eth0:10.0.0.8 pid=4102>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None pid=4068>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None,s2-eth5:None pid=4071>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=4074>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=4077>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=4080>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None,s6-eth3:None pid=4083>
<RemoteController c0: 192.168.7.106:6653 pid=4061>
mininet>
```

Figure 67 : affichage l'état du réseau simulé

➤ **Afficher flow table d'un switch**

Taper la command suivante: sudo ovs-ofctl dump-flows s1 sudo ovs-ofctl del-flows s1

```
mininet> sh ovs-ofctl dump-f flows s1
ovs-ofctl: unknown command 'dump-f'; use --help for help
mininet> sh ovs-ofctl dump-flows s1
cookie=0x2b000000000000000006, duration=1022.281s, table=0, n_packets=205, n_bytes=17425, priority=100,
dl_type=0x88cc actions=CONTROLLER:65535
cookie=0x2b00000000000001b, duration=1018.285s, table=0, n_packets=168, n_bytes=13328, priority=2,in
_port="s1-eth1" actions=CONTROLLER:65535
cookie=0x2b0000000000000006, duration=1022.281s, table=0, n_packets=0, n_bytes=0, priority=0 actions=
drop
mininet>
```

Figure 68 : affichage de la table des flux du switch 1

CHAPITRE 3.LA MISE EN PLACE ET RESULTATS

Tables de flux introduites Nous avons mis quelques configurations des tables de flux que nous avons créé et introduites dans les différents Switches de la topologie via le contrôleur SDN, décrivant leurs comportements en fonction des paquets reçu.

Statistiques du connecteur de nœud pour l'ID de nœud - openflow:1

ID du connecteur de nœud	Paquets Rx	Paquets d'émission	Octets de réception	Octets d'émission	Gouttes Rx	Gouttes d'émission	Erreurs Rx	Erreurs d'émission	Erreurs de trame Rx
flux ouvert : 1 : 1	635	479	52867	40715	0	0	0	0	0
flux ouvert : 1 : LOCAL	0	0	0	0	0	0	0	0	0

Figure 69 : Table de flux SW.OF.1

Statistiques du connecteur de nœud pour l'ID de nœud - openflow:2

ID du connecteur de nœud	Paquets Rx	Paquets d'émission	Octets de réception	Octets d'émission
flux ouvert : 2 : 1	518	674	44030	56182
flux ouvert : 2 : 2	572	656	47866	54530
flux ouvert : 2 : 3	572	648	47866	53746
flux ouvert : 2:4	572	648	47866	53746
flux ouvert : 2 : LOCAL	0	0	0	0
flux ouvert : 2:5	572	648	47866	53746

Figure 70 :Table de flux SW.OF.2

Statistiques du connecteur de nœud pour l'ID de nœud - openflow:3

ID du connecteur de nœud	Paquets Rx	Paquets d'émission	Octets de réception	Octets d'émission
flux ouvert : 3 : LOCAL	0	0	0	0
flux ouvert : 3 : 1	670	586	55720	49056
flux ouvert : 3 : 2	29	686	2002	57176
flux ouvert : 3 : 3	28	674	1960	56000

*Figure 71 : Table de flux SW.OF3***Statistiques du connecteur de nœud pour l'ID de nœud - openflow:4**

ID du connecteur de nœud	Paquets Rx	Paquets d'émission	Octets de réception	Octets d'émission
flux ouvert : 4 : 1	670	594	55616	49736
flux ouvert : 4 : 2	28	684	1960	56876
flux ouvert : 4:3	28	684	1960	56876
flux ouvert:4:LOCAL	0	0	0	0

Figure 72: Table de flux SW.OF4

Statistiques du connecteur de nœud pour l'ID de nœud - openflow:5

ID du connecteur de nœud	Paquets Rx	Paquets d'émission	Octets de réception	Octets d'émission
flux ouvert : 5 : 1	681	605	56551	50671
flux ouvert : 5 : 2	28	695	1960	57811
flux ouvert : 5 : 3	28	695	1960	57811
flux ouvert:5:LOCAL	0	0	0	0

*Figure 73 :: Table de flux SW.OF5***Statistiques du connecteur de nœud pour l'ID de nœud - openflow:6**

ID du connecteur de nœud	Paquets Rx	Paquets d'émission	Octets de réception	Octets d'émission
flux ouvert:6:LOCAL	0	0	0	0
flux ouvert : 6 : 1	689	613	57231	51351
flux ouvert : 6 : 2	28	701	1960	58295
flux ouvert : 6:3	28	703	1960	58491

Figure 74 : Table de flux SW.OF6

RESTCONF

La configuration et la programmation du réseau peut se faire directement en communiquant avec l'API RESTCONF. Pour accéder à l'explorateur REST à partir d'un navigateur et y poster des requêtes il faut installer les fonctionnalités suivantes :

```
opendaylight-user@root>feature:install odl-mdsal-all odl-restconf-all
```

Ensuite introduire l'url <AdresseIP> :8181/apidoc/explorer/index.html, qui nous réorientera vers la documentation complète de REST et toutes les interactions possibles avec l'API.

API	Date	Action	Action	Action	Action
XSQL	(2014-06-26)	Show/Hide	List Operations	Expand Operations	Raw
aaa-authn-model	(2014-10-29)	Show/Hide	List Operations	Expand Operations	Raw
config	(2013-04-05)	Show/Hide	List Operations	Expand Operations	Raw
flow-capable-transaction	(2015-03-04)	Show/Hide	List Operations	Expand Operations	Raw
flow-topology-discovery	(2013-08-19)	Show/Hide	List Operations	Expand Operations	Raw
ietf-interfaces	(2014-05-08)	Show/Hide	List Operations	Expand Operations	Raw
ietf-netconf	(2011-06-01)	Show/Hide	List Operations	Expand Operations	Raw
ietf-netconf-monitoring	(2010-10-04)	Show/Hide	List Operations	Expand Operations	Raw
kitchen-service-impl	(2014-01-31)	Show/Hide	List Operations	Expand Operations	Raw
lldp-speaker	(2014-10-23)	Show/Hide	List Operations	Expand Operations	Raw
nc-notifications	(2008-07-14)	Show/Hide	List Operations	Expand Operations	Raw

Figure 75 :Documentation du REST API

Chaque élément de cette liste dispose d'une panoplie d'instructions qui peut être explorée et utilisé par le biais de ce même explorateur.

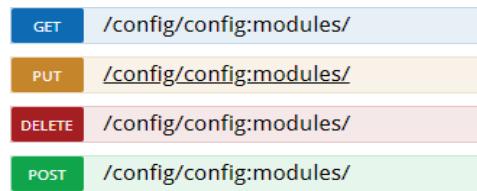


Figure 76: Types de requêtes REST API

Quatre types de requêtes sont supportés par l'API

➤ POSTMAN

Il est possible d'utiliser l'interface REST à partir d'un explorateur, mais on a trouvé plus pratique de l'exploiter avec l'outil POSTMAN disponible en extension du navigateur Google Chrome. Il offre une interface graphique intuitive et permet de sauvegarder les scripts dans des collections (Essaye, 2017). Voici une vue d'ensemble de cet outil dans la **figure 102**.

Figure 77: Vue d'ensemble de POSTMAN. L'image montre l'interface de l'outil POSTMAN. On peut voir une barre d'outils en haut, une liste de collections et environnements sur la gauche, et la fenêtre principale où un GET request est configuré pour l'URL http://192.168.7.106:8181/restconf/config/openaylight-inventory:nodes/node/openflow:1/table/0/flow/1. Le corps de la requête contient du XML. En bas, il y a des onglets pour Body, Cookies, Headers, Test Results et Visualize, avec Visualize actif. Une illustration d'un robot peignant sur une toile est visible au bas de l'écran.

Figure 77:Vue d'ensemble de POSTMAN

Pour interagir avec n'importe quelle interface REST 3 éléments doivent être fournis :

- **Les en têtes :** Pour déclarer le langage du script introduit (json ou xml)

et pour l'authentification.

- **L'Adresse :** Celle-ci dépend de l'opération voulue (POST, PUT, DELETE, GET), de l'identifiant de l'équipement et du flux.
- **Le Corps du script :** Pour les requêtes POST et PUT on doit introduire les données sous forme de code.

Aussi après chaque requête nous recevrons une réponse HTTP. Un code 200 veut dire quela requête a été effectuée avec succès, 400 et 500 par contre décrivent une erreur qui sera spécifié au pied de la page.

➤ Configurer l'authentification (protocoles SSH et SSL)

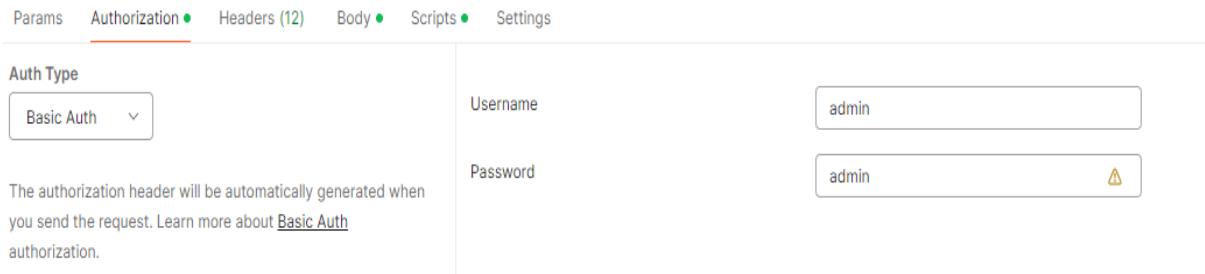


Figure 78: l'authentification

➤ A présent nous allons envoyer une requête PUT avec l'entête suivant



The screenshot shows a REST client interface with the following tabs: Params, Authorization, Headers (12), Body, Scripts, and Settings. The Headers tab is selected, indicated by a red underline. Below the tabs, it says "Headers ① 10 hidden". A table displays the header settings:

Key	Value	Description
<input checked="" type="checkbox"/> Content-Type	application/xml	
<input checked="" type="checkbox"/> Accept	application/xml	
Key	Value	Description

Figure 79: En tête d'une requête PUT

Ensuite nous allons introduire le code suivant écrit en XML, qui consiste en une simple opération de firewalling de couche 2, à l'adresse :

`http://192.168.7.106:8181/restconf/config/opendaylightinventory:nodes/node/openflow:1/table/0/flow/1` Est l'identifiant du switch, 0 le numéro de la table et 1 l'ID de l'entrée flux. La topologie du réseau est celle illustrée précédemment dans l'interface graphique du contrôleur.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <id>1</id> <!-- ID unique du flux, assurez-vous que cet ID est différent pour chaque flux -->
  <flow-name>s1</flow-name>
  <priority>10</priority>
  <idle-timeout>0</idle-timeout>
  <hard-timeout>0</hard-timeout>
  <match>
    <ipv4-destination>10.0.0.2/32</ipv4-destination>
    <ipv4-source>10.0.0.1/32</ipv4-source>
  </match>
  <instructions>
    <instruction>
      <order>0</order>
      <apply-actions>
        <action>
          <order>0</order>
          <output-action>
            <output-node-connector>1</output-node-connector>
          </output-action>
        </action>
      </apply-actions>
    </instruction>
  </instructions>
</flow>
```

Figure 80 Exemple de script XML

➤ **Résultat de la requête envoyer**

Nous recevrons une réponse HTTP. Un code 200 veut dire quela requête a été effectuée avec succès

⊕ Status: 200 OK Time: 245 ms Size: 769 B  Save as example

Figure 81 : Résultat de la requête envoyer

3.5. Conclusion

Dans ce chapitre nous avons décrit les opérations d'administration les plus fréquentes. En plus de la simplicité de l'implémentation et le gain de temps considérable en ce qui concerne la programmation du comportement des réseaux ce qui nous rend beaucoup moins limités en termes de réalisation de topologies complexes comme dans le cas de réseau LAN. Les résultats ont démontré la fiabilité de l'architecture en termes du respect des politiques prédéfinis. Un test d'accessibilité a montré le comportement du réseau en termes de restrictions, celui-ci a confirmé la conformité des communications envers ce qui a été préétabli.

Conclusion générale

Ce mémoire a mis en lumière l'importance et les avantages de l'adoption du SDN au sein de Bejaia Logistique. La flexibilité, la scalabilité et la centralisation offertes par cette technologie en font une solution idéale pour surmonter les limitations des réseaux traditionnels. À travers l'analyse théorique et les simulations pratiques, nous avons démontré comment le SDN peut transformer les infrastructures réseau.

Notre étude a non seulement confirmé les bénéfices du SDN pour la gestion réseau, mais a également souligné les défis techniques et opérationnels à surmonter pour une implémentation réussie. En particulier, l'importance d'une stratégie de migration bien planifiée et d'une formation adéquate pour le personnel a été mise en évidence.

En conclusion, le SDN représente une avancée majeure dans le domaine des réseaux informatiques, offrant une réponse adéquate aux exigences croissantes de modernité et de performance. Ce travail constitue une contribution significative à l'innovation technologique au sein de Bejaia Logistique, ouvrant la voie à des réseaux plus intelligents et adaptatifs. Les perspectives futures incluent l'intégration de nouvelles technologies émergentes et l'optimisation continue des performances réseau grâce à des recherches et des développements continus dans le domaine du SDN.

Bibliographie

- [1] Ifri. "Nos Filiales." Bejaia, Algérie. 2024. Consulté le 7 juin 2024. Disponible à : <https://ifri-dz.com/nos-filiales>.
- [2] G. Pujolle, Les réseaux. Editions Eyrolles, 2014.
- [3] Amazon Web Services. "Qu'est-ce qu'une mise en réseau informatique ?" 2024. Consulté le 7 juin 2024. Disponible à : <https://aws.amazon.com/fr/what-is/computer-networking/>
- [4] École La Mâche. "Topologie des Réseaux Informatiques." 2024. Consulté le 7 juin 2024. Disponible à : <https://sti2d.ecolelamache.org/iirseauxinformatiques7topologiedesrseaux.html>
- [5] A. Johnson, "Scaling Networks v6 Companion Guide and Lab ValuePack," Indianap. U.S. Am., 2017.
- [6] École La Mâche. "Le Modèle de Référence OSI." 2024. Consulté le 7 juin 2024. Disponible à : <https://sti2d.ecolelamache.org/iirseauxinformatiques4lemodlederfrenceosi.html>
- [7] Wikipédia. "Adresse IP." 2024. Consulté le 7 juin 2024. Disponible à : https://fr.wikipedia.org/wiki/Adresse_IP
- [8] Z. Mammeri, "Réseaux sans fils Caractéristiques et principaux standards," M1 InfoCours Réseaux IRIT Univ. Paul Sabatier Toulouse Httpwww Irit Fr~ Zoubir MammeriChap6WLAN Pdf.
- [9] A. S. Tanenbaum, D. Wetherall, M.-C. Baland, E. Burr-Campillo, and S. Pauquet, Réseaux, vol. 4. Pearson Education, 2003.
- [10] A. S. Tanenbaum, D. Wetherall, M.-C. Baland, E. Burr-Campillo, and S. Pauquet, Réseaux, vol. 4. Pearson Education, 2003.
- [11] Cours Travaux Pratiques : Administration des Réseaux Préparé par Prof. Mohand YAZID, 2022/2023

Bibliographie

- [12] Cloudflare. "Internet Control Message Protocol (ICMP)" Glossary, Cloudflare Learning Center. Consulté le 7 juin 2024. [En ligne]. Disponible à l'adresse : <https://www.cloudflare.com/fr-fr/learning/ddos/glossary/internet-control-message-protocol-icmp/>.
- [13] Huawei. "Forum Huawei", Forum Huawei. Consulté le 7 juin 2024. [En ligne]. Disponible à l'adresse : <https://forum.huawei.com/enterprise/fr/diff>.
- [14] A. El Bardai, "Virtualisation d'une plateforme de gestion de contexte," masters, École de technologie supérieure, Montréal, 2015
- [15] P. Primet, O. Mornard, and J.-P. Gelas, "Évaluation des performances réseau dans le contexte de la virtualisation XEN," INRIA, report, 2007. Accessed: Jul. 08, 2020. [Online]. Available: <https://hal.inria.fr/inria-00203461>.
- [16] N. Triki, "Gestion de ressources dans les infrastructures de virtualisation de réseaux," masters, École de technologie supérieure, 2013.
- [17] Benamrane, F. (2017) Etude des Performances des Architectures du Plan de Contrôle des Réseaux 'Software-Defined Networks (Thèse de doctorat). Université de Rabat
- [18] Badotra, S., & Singh, J. (2017). OpenDaylight as a Controller for Software Defined Networking. International Journal of Advanced Research in Computer Science, 8(5), 1105-1111.
- [19] Jaouad, Saa'd. "PROJET SDN", INSTITUT SPECIALISE DE TECHNOLOGIE APPLIQUEE NTIC BENI MELLAL. Consulté le 7 juin 2024.
- [20] M. Jaafreh, "Toward integrating software defined networks with the Internet of Things: a review". In : Cluster Computing, p. 1-18, 2021.
- [21] Azodomolky, S. (2013). Software Defined Networking with Openflow. Packt Publishing. Première édition. Birmingham, Royaume uni.
- [22] « Software-Defined Networking (SDN) Definition », Open Networking Foundation. [En ligne]. Disponiblesur: <https://www.opennetworking.org/sdn-definition/>. [Consulté le: 19-nov-2018].

Bibliographie

- [23] Merabtine, Saadi. "SDN IOT", Université de Béjaia. Consulté le 7 juin 2024.
- [24] Takayuki Sasaki et al. "SDNsec : Forwarding accountability for the SDN data plane". In : 2016 25th International Conference on Computer Communication and Networks (ICCCN). IEEE. 2016, p. 1-10.
- [25] Ihssane Choukri, Mohammed Ouzzif et Khalid Bouragba. "Software Defined Networking (SDN) : Etat de L'art". In : Colloque sur les Objets et systèmes Connectés. 2019.
- [26] Yvon Zafimahefa Andrianirina. "Développement dun réseau défini par logiciel (SDN) programmable,transparent et ouvert". In : (Thèse de doctorat) Université du Québec en Outaouais (2021).
- [27] Vivek Tiwari. "SDN and OpenFlow for beginners with hands on labs". In : MMDD Multimedia LLC., Kindle Edition, Northville (2013).
- [28] "Mise en place d'un réseau SDN", by AMEL REBOUCHE. Consulté le 7 juin 2024.
- [29] "Mise en place d'un réseau SDN", by AMEL REBOUCHE. Consulté le 7 juin 2024
- [30] "Partie 2 - CH1 : Définir la technologie SDN", Scribd. Consulté le 7 juin 2024. [En ligne]. Disponible à l'adresse : <https://www.scribd.com/document/654703180/Partie-2-CH1-Definir-la-technologie-SDN-1>
- [31] Mohamed BARHOUMI,Haithem BELHADEF,Maha MEZZI,2021-2022,LES RÉSEAUX SDNS, Université de Tunis El Manar Institut Supérieur d'Informatique
- [32] "Orchestration réseau", LeMagIT. Consulté le 7 juin 2024. [En ligne]. Disponible à l'adresse : <https://www.lemagit.fr/definition/Orchestration-reseau>.
- [33] "Mise en place d'un réseau SDN", by AMEL REBOUCHE. Consulté le 7 juin 2024.
- [34] Jaouad, Saa'd. "Etude et implémentation d'une architecture SDN, LAN", Scribd. Consulté le 7 juin 2024.
- [35] <https://www.sdxcentral.com/articles/news/openflow-2-0-brings-new-flexibility-to-switches/2014/03/>

Bibliographie

- [36] Goransson, P., Black, C., & Culver, T. (2016). Software Defined Networks a Comprehensive Approach. 2éme édition. New York, USA. : Elsevier Science.
- [37] E. Cancerès, Le Protocole OpenFlow dans l'Architecture SDN. Consulté sur:
http://www.efort.com/r_tutoriels/OPENFLOW_EFORT.pdf, 2016.
- [38] B. Fouad and M. Mouad, “Etude des Performances des Architectures du Plan de Contrôle des Réseaux ‘Software- Defined Networks.’” Jan. 2017, [Online]. Available: <https://www.researchgate.net/publication/316620880>.
- [39] E. Cancerès, *Le Protocole OpenFlow dans l'Architecture SDN*. Consulté sur:
http://www.efort.com/r_tutoriels/OPENFLOW_EFORT.pdf, 2016.
- [40] H. BOUIDA née SAIDI, “Etude et mise en oeuvre d'une solution SDN .,” Thesis, 11-03-2018, 2017.
- [41] T. Issa, K. B. Médard, and A. Ferdinand, “Etude du nomadisme dans un Cloud éducatif administré par la technologie SDN/OpenFlow,” 2016
- [42] D. V. Bernardo and B. B. Chua, “Introduction and analysis of SDN and NFV security architecture (SN-SECA),” in 2015 IEEE 29th international conference on advanced
- [43] M. Amadou Diouhé DIALLO, ”SDN : Etat de l’art et perspectives pour une solution contre le DoS” , université Assane SECK de ziguinchor , 2018/2019

Résumé : Avec l'évolution rapide des technologies et des besoins en connectivité, les réseaux traditionnels montrent leurs limites. Pour y remédier, de nombreuses entreprises se tournent vers les réseaux SDN (Software-Defined Networking), qui permettent une gestion centralisée et une meilleure programmabilité des infrastructures. Cette étude vise à concevoir et mettre en place un réseau SDN pour Bejaia Logistique. La question centrale est de savoir comment un réseau SDN peut améliorer la gestion et la performance des réseaux de l'entreprise. Une analyse de l'infrastructure actuelle a révélé ses limitations et besoins d'amélioration. Une étude des concepts SDN, incluant contrôleurs, commutateurs et protocoles comme OpenFlow, a été menée. Sur cette base, une architecture SDN adaptée a été proposée et mise en œuvre avec des outils comme VMware Workstation et Mininet. Les résultats montrent que le SDN offre une flexibilité accrue, une gestion unifiée améliorée, et une optimisation des réseaux informatiques, répondant efficacement aux besoins de connectivité et de gestion des données de l'entreprise.

Mots clés – SDN, OPENFLOW, OVSWITCH

Abstract: With the rapid evolution of technology and increasing connectivity and data management needs, traditional networks show their limitations in terms of flexibility and efficiency. To address these challenges, many companies are exploring the implementation of SDN (Software-Defined Networking) networks, which allow for centralized management and better programmability of network infrastructures. This study aims to design and implement an SDN network for Bejaia Logistics. The main issue addressed is how the implementation of an SDN network can improve network management and performance within the company. A detailed analysis of the company's current infrastructure was conducted, highlighting its limitations and areas for improvement. An in-depth study of SDN concepts and principles was also carried out, including essential components such as SDN controllers and switches, as well as protocols like OpenFlow. Based on these analyses, an SDN architecture tailored to the company was proposed. Practical implementation was carried out using tools like VMware Workstation and Mininet, with detailed procedures for installation and configuration provided. The results of this implementation show that the SDN network offers greater flexibility, improved unified management, and optimization of IT networks, allowing the company to effectively meet its growing connectivity and data management needs. These findings suggest that adopting SDN networks could be a viable and advantageous solution for many companies seeking enhanced network management and performance.

Key-words- SDN, OPENFLOW, OVSWITCH

