République Algérienne Démocratique Et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Kasdi Merbah-Ouargla

Faculté des Nouvelles Technologies de l'Information et de la Communication

Département d'Informatique et Technologie de l'information

Mémoire Master Professionnel

Domaine: Informatique et Technologie de l'Information

Filière: Informatique

Spécialité: administration et sécurité des réseaux

Présentée par : TIDJANI NESRINE ET TIDJANI FATMA ZOHRA

Thème

Evaluation de performances des protocoles OSPF et Open Flow

Mémoire soutenu le: 15 juin 2023 devant le jury composé de:

NOM et Prénom	Grade	Université	
Mr. KHALDI Amine	M.C.A	UKM Ouargla	Président
Mr. KAHLESSENANE Fares	M.A.A	UKM Ouargla	Rapporteur
Mr. MECHALIKH Charafeddine	M.C.B	UKM Ouargla	Examinateur

Année universitaire: 2022 /2023

Remerciements

Nous remercions « Allah le tout puissant », qui nous a donné la force le courage, la volonté et la patience pour l'accomplissement de ce travail.

Nous exprimons nos remerciements et notre profonde gratitude à toute personne ayant contribué de près ou de loin à l'élaboration de ce projet par une quelconque forme de participation, trouve ici le témoignage de notre plus profonde reconnaissance.

De prime abord, nos hommages, nos respectueux remerciements et nos reconnaissances à notre encadreur Monsieur Fares Kahlessenane pour le temps réservé à l'étude de notre projet son suivi, ces conseils, aux jugements et précieuses recommandations tout au long de cette période.

Nos remerciements à Messieurs Mahdjoub Bachir et Abdellatif Cheradid pour leur précieux conseils et contribution.

Enfin, nous remercions tous les enseignants de l'université Kasdi Merbah- Ouargla, qui ont contribué à notre Formation.

Bien entendu, Nous tenons surtout á remercié nos parents: Tidjani Mohammed, et Tidjani Sadok, pour leurs sacrifices, leurs patiences et leurs soutiens, tout au long de leurs vies.

Dédicace

- A ma chère mère,
- A mon cher papa,
- A mes Grands Parents,
- A mon cher frère « Mohamed Sofiane » et ma chère sœur « Fatima Zahra - Zola», et « Mimi »
- A mes oncles et tantes,
- A toute la famille TIDJANI et YOUMBAI

Tidjani Nesrine

- A ma chère mère,
- A mon cher père,
- A mes chères tantes : Aicha, Bahidja, Professeur Thouria, Fatiha, Zohra……
- A mes chers frères Bachir et Laid,
- A ma chère sœur Ouahiba,
- A mes oncles : Hadj et Sif,
- A toute la famille TIDJANI

Tidjani Fatima Zohra

Liste des figures

Liste des tableaux

Table des matières

Liste des abréviations

SDN: Software-defined networking

REST: Representational State Transfer

JSON: JavaScript Object Notation

XML: Extensible Markup Language

OSPF: Open Shortest Path First

BGP: Border Gateway Protocol

RIP: Routing Information Protocol

EIGRP: Enhanced Interior Gateway Routing Protocol

IGP: Interior Gateway Protocol

EGP: Exterior Gateway Protocol

ICMP: Internet Control Message Protocol

MLPS: Multiprotocol Label Switching

IP: Internet Protocol

MAC: Media Access Control

ONF: l'Open Networking Foundation

CAMs: Content-Addressable Memories

TCAMs: Tenary contenant addressables memory

API: Application Programming Interface

CLI: Command Line Interface

AS: systems autonomes

RFC: Request for Comments

DUAL Diffusing Update Algorithm

LSAL: Link State Advertisement

HTTP: Hyper Text Transfer protocol

LSDB: Link State Database

SPF: Shortest Path First

IOS: International Organization for Standardization

VDI: Virtual Desktop Infrastructure

RAM: Read Access memory

API: interface de programmation d'application

TLS : Transport Layer Security

VM: Virtual Machine

TCP: Transmission Control Protocol

Résumé

Ce projet a pour objectif d'évaluer les performances du routage traditionnel et du SDN dans des scénarios de réseau évolutif. Il utilise des simulations de réseau pour modéliser le comportement du réseau dans des conditions contrôlées et obtenir des mesures précises des performances. Les résultats obtenus seront ensuite comparés afin de déterminer la méthode la plus adaptée pour un réseau donné, l'étude comparative des méthodes de routage traditionnel et SDN met en évidence les avantages et les inconvénients de chaque approche. Le routage traditionnel, basé sur des protocoles décentralisés, offre une stabilité, une fiabilité et une faible latence. Cependant, il peut être complexe à configurer et manquer de flexibilité dans les environnements en évolution rapide. En revanche, le SDN introduit une approche centralisée qui permet une gestion plus flexible, une programmabilité accrue et une adaptation rapide aux changements du réseau. Bien qu'il soit plus facile à configurer et à gérer, il peut présenter une légère augmentation de la latence en raison de la centralisation des décisions de routage. Le choix entre ces méthodes dépendra des besoins spécifiques du réseau, de la flexibilité requise, de la complexité acceptable et des performances souhaitées. En conclusion, ce projet utilise des simulations de réseau pour comparer les performances de l'OSPF et du SDN dans des scénarios de réseau évolutif. Les résultats obtenus dans notre projet, affichent pour le SDN un taux de 85%, tandis que l'OSPF présente un pourcentage de 69,4 %. Le SDN offre des avantages significatifs en termes de flexibilité et de gestion simplifiée.

Mots-clés: SDN, routage traditionnel, performances, réseau.

Abstract

This project aims to evaluate the performance of traditional routing and SDN (Software-Defined Networking) in scalable network scenarios. It utilizes network simulations to model the network behavior under controlled conditions and obtain precise performance measurements. The obtained results will then be compared to determine the most suitable method for a given network. The comparative study of traditional routing and SDN methods highlights the advantages and disadvantages of each approach.Traditional routing, based on decentralized protocols, offers stability, reliability, and low latency. However, it can be complex to configure and lack flexibility in rapidly evolving environments. On the other hand, SDN introduces a centralized approach that allows for more flexible management, increased programmability, and rapid adaptation to network changes. While it is easier to configure and manage, it may result in a slight increase in latency due to the centralization of routing decisions.The choice between these methods will depend on the specific needs of the network, required flexibility, acceptable complexity, and desired performance. In conclusion, this project utilizes network simulations to compare the performance of OSPF (Open Shortest Path First) and SDN in scalable network scenarios. The results obtained in our project show that SDN achieves a rate of 85%, while OSPF has a percentage of 69.4%. SDN offers significant advantages in terms of flexibility and simplified management.

Keywords: SDN, Traditional routing, performance, Network

ملخص

يهدف هذا المشروع إلى تقييم أداء التوجيه التقليدي و SDN في سيناريوهات الشبكة القابلة للتطوير. يستخدم محاكاة الشبكة لنمذجة سلوك الشبكة في ظل ظروف خاضعة للرقابة والحصول على قياسات أداء دقيقة. سيتم بعد ذلك مقارنة النتائج التي تم الحصول عليها من أجل تحديد الطريقة الأنسب لشبكة معينة ، وتسلط الدراسة المقارنة لطرق التوجيه التقليدية و SDN الضوء على مزايا وعيوب كل نهج. يوفر التوجيه التقليدي ، المستند إلى البروتوكولات اللامركزية ، الاستقرار والموثوقية وزمن انتقال منخفض. ومع ذلك ، قد يكون تكوينه معقدًا ويفتقر إلى المرونة في البيئات سريعة التغير. في المقابل ، يقدم SDNنهجًا مركزيًا يسمح بإدارة أكثر مرونة ، وزيادة قابلية البرمجة ، والتكيف السريع مع تغييرات الشبكة. على الرغم من أنه من الأسهل تكوينه وإدارته ، فقد يواجه زيادة طفيفة في زمن الوصول بسبب قرارات التوجيه المركزية. سيعتمد الاختيار بين هذه الأساليب على الاحتياجات المحددة للشبكة والمرونة المطلوبة والتعقيد المقبول والأداء المطلوب. في الختام ، يستخدم هذا المشروع محاكاة الشبكة لمقارنة أداء OSPF و SDN في سيناريوهات الشبكة القابلة للتطوير. النتائج التي تم الحصول عليها في مشروعنا ، تظهر لـ SDN معدل 85٪ ، في حين أن OSPF تقدم نسبة 69.4٪. تقدم SDN مزايا كبيرة من حيث المرونة والإدارة المبسطة.

الكلمات المفتاحيةّ: شبكات التحكم بالبرمجيات، التوجيه التقليدي، الأداء، الشبكة

Introduction Générale

Les réseaux informatiques sont essentiels à notre société moderne pour connecter les individus, les organisations et les systèmes à travers le globe. Le routage est un élément essentiel du mécanisme central de transmission des données dans ces réseaux. Au fil du temps, diverses approches de routage ont été développées, y compris le routage conventionnel basé sur des protocoles de routage distribués et le SDN (Software Defined Networking), qui utilise une approche centralisée.

Dans un contexte d'évolution continue des réseaux, il est primordial de comprendre et d'évaluer les différentes approches de routage disponibles. La SDN, en tant qu'innovation récente, promet une gestion plus flexible et des performances réseau optimisées. Cependant, il est essentiel de comparer le SDN et les méthodes de routage traditionnelles afin de déterminer les avantages, les limites, les performances et l'impact de chaque approche.

L'objectif principal de cette étude est de mener une analyse comparative approfondie des méthodes de routage du SDN et traditionnelles. Nous voulons décrire les avantages, les inconvénients, les performances et l'impact de chaque méthode dans le contexte actuel des réseaux de communication. Cette recherche vise à fournir des informations utiles aux chercheurs, aux ingénieurs et aux décideurs travaillant dans le domaine des TIC.

Ce travail est divisé en différents chapitres. Le premier chapitre offre un aperçu des réseaux définis par le logiciel SDN. Nous examinerons les principales idées, structures et fonctionnalités du SDN, mettant en évidence ses avantages potentiels par rapport aux techniques de routage conventionnelles.

Le deuxième chapitre met l'accent sur les recherches antérieures et actuelles et présente une revue de la littérature sur les méthodes de routage traditionnelles. Nous examinerons les fonctionnalités, les performances et les limites des protocoles de routage distribué les plus courants.

Une méthodologie détaillée pour mener des études comparatives entre le SDN et l'OSPF sera présentée dans le troisième chapitre. Pour évaluer et comparer ces approches, nous proposerons les mesures de performance pertinentes, des scénarios d'utilisation spécifiques et des critères d'évaluation. Nous présenterons les résultats et les conclusions de notre étude dans ce chapitre. Nous discuterons également de leurs implications pour les réseaux de communication modernes.

Ce mémoire s'achève par une conclusion générale et les perspectives de notre travail:

Chapitre1: software – defined networking

1 Introduction

Par rapport aux autres technologies de télécommunication, et pendant longtemps, les technologies réseau ont évolué à un rythme très faible. Les équipements de réseau tels que les commutateurs et les routeurs ont été développés par les fabricants avec des plans de contrôle et de données étroitement liés.

Chaque fournisseur conçoit son propre firmware pour exploiter son matériel de manière exclusive et privée. Le développement, l'évaluation et le déploiement d'un nouveau protocole peut prendre de 5 à 10 années [1], Ce qui a ralenti l'avancement des innovations, et provoqué une augmentation considérable de la complexité, des coûts de gestion et de l'exploitation dans les technologies réseaux à chaque fois que de nouveaux services devaient être déployés dans les réseaux existants. L'existence de ces limitations a poussé à la collaboration des chercheurs et des leaders du marché industriel afin de repenser la conception des réseaux traditionnels. Dans ce contexte et suite à l'augmentation du nombre d'utilisateurs et des équipements, qui nécessitent un traitement distribué et l'adoption de l'information par plusieurs entreprises, ont induit le besoin d'avoir des réseaux dynamiques qui offrent la possibilité de s'adapter rapidement aux changements des requis. Cela a conduit à la conception de réseaux programmés par logiciel (SDN, Software-Defined Networking).

SDN n'est pas une proposition révolutionnaire, mais un mélange de propositions antérieures, notamment les réseaux programmables et la séparation du plan de contrôle et de données [2]. C'est le résultat d'un processus à long terme déclenché par le désir de porter le réseau « out of the box ».L'idée principale de SDN, et en remplacement de 4 D [3] et Ethane [4], consiste à déplacer le plan de contrôle à l'extérieur des périphériques réseau et à laisser seulement le plan de données à l'intérieur. Le plan de contrôle est pris en charge par une application appelée "Contrôleur". Les périphériques réseau deviennent des équipements de transmission simples qui peuvent être programmés par le contrôleur. En effet, SDN permet principalement de centraliser la logique déterminant les politiques de gestion d'un réseau dans une ou plusieurs unités appelées contrôleurs. Ces contrôleurs communiquent avec le reste des équipements du réseau [5]. De ce fait, la définition d'une politique de gestion d'un réseau revient à écrire des programmes et les déployer dans les contrôleurs. Dans la plupart des cas, ces programmes seront compilés, en tenant compte de la topologie et des ressources disponibles, afin de générer les configurations nécessaires à chaque équipement du réseau pour mettre en œuvre les politiques désirées.

2 Définition SOFTWARE – DEFINED NETWORKING

Software-Defined Networking (SDN) Le réseau défini par logiciel (SDN) est la séparation par excellence du plan de contrôle du réseau du plan de transfert, où un seul plan de contrôle gère plusieurs appareils.

Cette séparation du plan de contrôle et du plan de données est définie comme interface de programmation d'application (API) entre le périphérique réseau et le contrôleur SDN. Le protocole OpenFlow [6] est un exemple d'API. Un commutateur avec interface programmable permet au contrôleur de communiquer et de définir des règles sur le commutateur. Le commutateur OpenFlow peut se comporter comme un routeur, un commutateur, un pare-feu ou un traducteur d'adresses réseau [7].

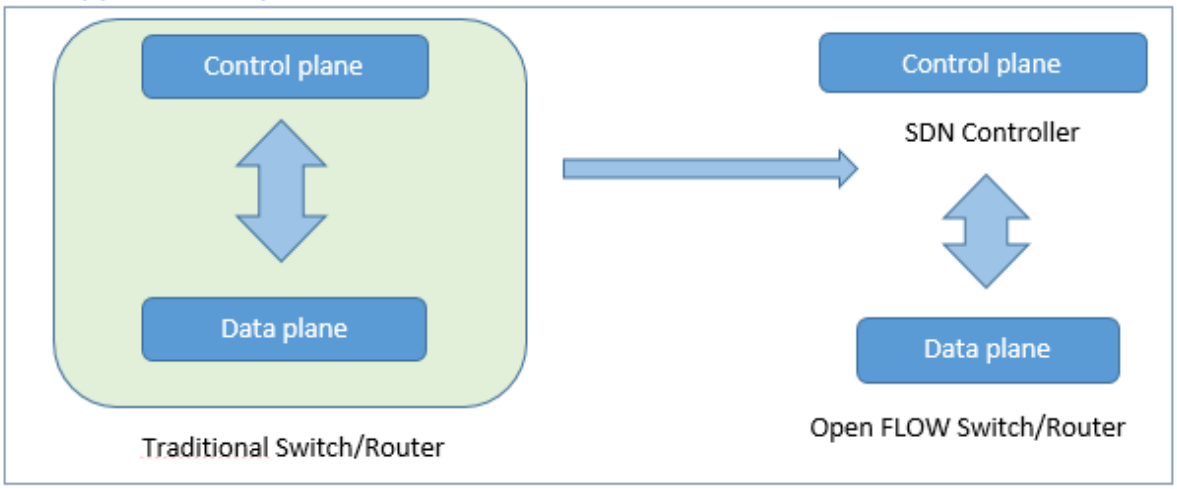


Figure 1: Form traditionnel network to SDN

3 Architecture de SDN :

3.1 Application Layer

Cette couche est constituée d'applications destinées aux utilisateurs finaux qui seront les consommateurs des services de communication SDN.

Les utilisateurs finaux utilisent les services de communication SDN via l'API Northbound tels que REST, JSON, XML, entre autres.

Cette API nord est utilisée pour connecter le contrôleur SDN aux services et applications ci-dessus, elle permet aux services et applications de simplifier et d'automatiser les tâches de configuration, de provisionnement et de gestion de nouveaux services dans le réseau, offrant

ainsi aux opérateurs de nouvelles façons de différenciation et d'innovation, ainsi que de répondre aux besoins des différentes applications grâce à la programmabilité du réseau SDN.
L'API nord traverse la limite entre cette couche et la couche de contrôle. Ce sont les interfaces les plus critiques, comme mentionné ci-dessus, elles prennent en charge un grand nombre d'applications et de services critiques pour l'utilisateur final.

3.2 Control Layer

Cette couche fournit une fonctionnalité de contrôle centralisée qui surveille le comportement du réseau de données via une interface ouverte.
Elle permet aux développeurs d'applications d'utiliser les capacités du réseau indépendamment de sa topologie ou de ses fonctions.
Dans cette couche, il faut mentionner le contrôleur SDN, car il s'agit de l'entité de contrôle logique responsable de la traduction des demandes de service SDN vers les chemins de données inférieurs, donnant ainsi à la couche application une vue abstraite du réseau à travers les statistiques et les événements possibles.
On pourrait dire que les contrôleurs sont le cerveau de cette architecture réseau, car ils ont le contrôle exclusif sur la manière de gérer et de configurer les nœuds du réseau pour diriger correctement les flux de trafic.
De plus, l'architecture leur permet de générer une large gamme de ressources de plan de données, leur offrant ainsi la possibilité d'unifier et de simplifier sa configuration.
Ils fournissent un ensemble d'API communes à la couche application (API nord), tout en implémentant un ou plusieurs protocoles réseau pour contrôler les périphériques réseau (interface sud).
SDN ne prend pas seulement en charge la mise en réseau orientée SDN - Protocoles de pointe - en fait, il prend également en charge les protocoles de réseau traditionnels tels qu'Open Shortest Path First (OSPF), MultiProtocol Label Switching (MLPS) ou Border Gateway Protocol (BGP).
Le contrôleur peut également comprendre un ensemble de modules qui lui permettent d'effectuer un ensemble de tâches du réseau de base : inventaire des appareils connectés, gestion des statistiques et autres fonctions. Les fournisseurs peuvent ajouter de nouvelles fonctionnalités dans le cœur du contrôleur en fonction de leurs besoins, étant l'un des points clés de l'architecture SDN.

3.3 Couche d'infrastructure

Cette couche est constituée des nœuds du réseau qui effectuent la commutation et le routage des paquets. Elle fournit un accès ouvert programmable via l'API sud, comme Open Flow.
Les API Southbound facilitent un contrôle efficace sur le réseau et permettent au SDN Controller d'apporter des modifications de manière dynamique en fonction des demandes et des besoins en temps réel.

Open Flow, qui a été développé par l'Open Networking Fondation (ONF), est la première et probablement la plus connue des interfaces sud. Il s'agit d'une norme de l'industrie qui définit la manière dont le contrôleur SDN doit interagir avec le plan de transfert pour apporter des ajustements au réseau, afin qu'il puisse mieux s'adapter à l'évolution des besoins de l'entreprise. Avec Open Flow, des entrées peuvent être ajoutées et supprimées de la table de flux interne des commutateurs et potentiellement des routeurs pour rendre le réseau plus réactif aux demandes de trafic en temps réel. Outre Open Flow, Cisco OpFlex (la réponse de l'entreprise à Open Flow) EST également une API sud bien connue.

3.4 Est/Ouest

Les interfaces côté Est/Ouest sont des interfaces de communication qui permettent généralement la communication entre les contrôleurs dans une architecture multi-contrôleurs pour synchroniser les états du réseau [8]. Ces architectures sont très récentes et aucun standard de communication inter-contrôleurs n'est actuellement disponible.

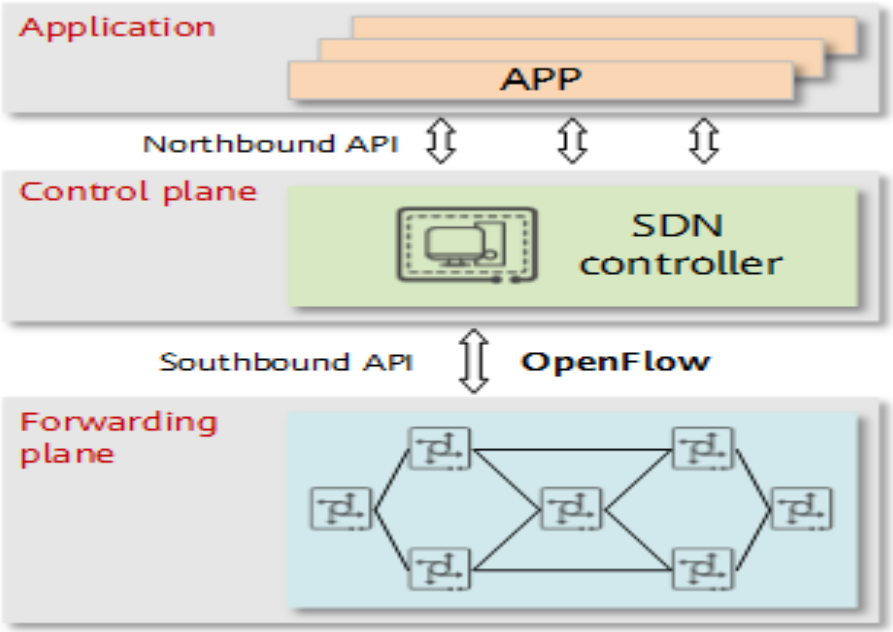


Figure 2:Interfaces de Communication Est/Ouest

4 Types de contrôleurs SDN

Beacon [9]: Beacon est un contrôleur SDN qui a été introduit en 2010. Il a été utilisé dans plusieurs projets de recherche. C'est un contrôleur basé sur Java. Il peut fonctionner sûr de Nombreuses plates-formes, y compris Linux multicœurs haut de gamme, et les téléphones Android.

DISCO [10] : Disco est un contrôleur distribué. Il est principalement utilisé pour les réseaux WAN et les réseaux superposés. Chaque contrôleur est responsable d'un domaine réseau. Les régulateurs communiquent entre eux par l'intermédiaire d'un canal inter-contrôleur. DISCO peut s'adapter dynamiquement à différentes topologies de réseaux hétérogènes.

IRIS [11]: IRIS est une plate-forme de contrôleur SDN pour contrôler la base Open Flow. Il peut gérer un grand réseau. IRIS prend en charge les architectures qui sont évolutives horizontalement. Ainsi, les serveurs peuvent être ajoutés dynamiquement au cluster de contrôleurs. Cela augmente le facteur de performance du contrôle plane.

Maestro [12] : Maestro est le premier système de contrôle Open Flow qui exploite le parallélisme. Dans Maestro, les programmeurs peuvent modifier la fonctionnalité des data plane en écrivant des programmes simples à filetage unique. Maestro à son propre ensemble des conceptions et des Techniques qui aident le protocole Open Flow. Il s'agit d'un contrôleur basé sur Java et très portable pour différents systèmes d'exploitation et architectures.

OpenDaylight [13]: OpenDaylight inspire de Beacon. Il s'agit d'un contrôleur basé de Java Dérivé de Beacon. Il supporte Open Flow et d'autres Southbound APIs. L'OpenDaylight est présent dans sa propre machine virtuelle Java (JVM).

NOX [14]: NOX est la première plate-forme SDN Open Flow Controller pour la construction D'applications de contrôle réseau. Il a été initialement développé par Nicira Networks, aux côtés d'Open Flow. Plus tard, NOX a été donné à la communauté SDN. Les applications peuvent être en Python ou en C++ et peuvent être chargées dynamiquement.

POX [15] : Pox est similaire au NOX Controller. POX est un contrôleur SDN qui permet le développement et le prototypage rapide du réseau. Il suit le protocole Open Flow, et qui sert a joué le rôle d'un framework entre les commutateurs Open Flow.

Floodlight [16]: Floodlight est un contrôleur SDN Open source. Il s'agit d'un contrôleur Open Flow de classe entreprise, basé sur Java. Il fonctionne à la fois avec les commutateurs Physiques et les commutateurs virtuels qui utilisent le protocole Open Flow.

Ryu [17]: Ryu est un contrôleur SDN basé sur des composants. Ryu signifie "flux" en japonais. Ryu fournit de nombreux composants logiciels et API étendus pour créer et gérer les applications réseau. Ryu supporte les protocoles comme Open Flow, Netconf, OF-config, etc. Ryu est complètement implémenté en langage Python. Il est sous licence Apache 2.0.

Controller	Architecture	Open flow	Language	Tolérance pannes	API Nord
Beacon	Centralisé multi-thread	V 1.0	Java	NON	Ad-hoc API
Floodlight	Centralisé multi-threadDistribué	V 1.1 1.2 1.3 1.4	Java	NON	RESTful API
POX	Centralisé	V 1.0	Python	NON	Ad-hoc API
OpenDaylight	Distribué	V 1.0 1.3	Java	NON	REST/RESTCONF
Ryu	Centralisé	V 1.1 1.2 1.3 1.4 1.5	Python	NON	Ad-hoc API
NOX	Centralisé	V 1.0 1.1 1.2 1.3	C++	NON	Ad-hoc API
DISCO	Distribué	v1.1	Java	OUI	RESTful API
IRIS	Centralisé	/	Java	/	RESTful API
Maestro	Centralisé multi-threaded	v1.0	Java	NON	Ad-hoc API

Tableau 1:Caractéristiques des contrôleurs SDN

5 Les type switch

5.1 Switch SDN logiciel

C'est le moyen le plus simple pour créer un équipement SDN. Les tables de flux, les entrées et les champs de correspondance sont facilement implémentés dans les structures de données d'un software. Ce modèle est plus lent et moins efficace, car il ne bénéficie pas d'une accélération hardware, mais présente l'avantage d'être plus flexible, plus riche dans les actions disponibles, et supporte un nombre d'entrées beaucoup plus important. Les switches logiciels sont très présents dans les environnements virtualités et sont généralement open source. Les deux principales alternatives sont Open vSwitch (OVS) de Nicira et Indigo de Big Switch.

5.2 Switch SDN matériel

Ces implémentations sont plus rapides, et sont la seule possibilité pour un environnement très haut débit (100Gbs) étant sensible aux performances. Pour transcrire les différentes entrées de flux et leurs composantes dans les switchs, on a opté pour l'utilisation de hardware spécialisé, pour le traitement de couche 2 (MAC) on utilise les CAMs (Content-Addressable Memories) et

pour la couche 3 les TCAMs (Ternary Content-addressable Memories). Ces switchs sont adaptés au data center et au cœur du réseau, la politique de flux n'y sont pas centrés sur les utilisateurs ce qui fait que le nombre d'entier y est moins important que dans les switchs qui sont plus près de l'accès Les mode opérationnel de contrôleur [18].

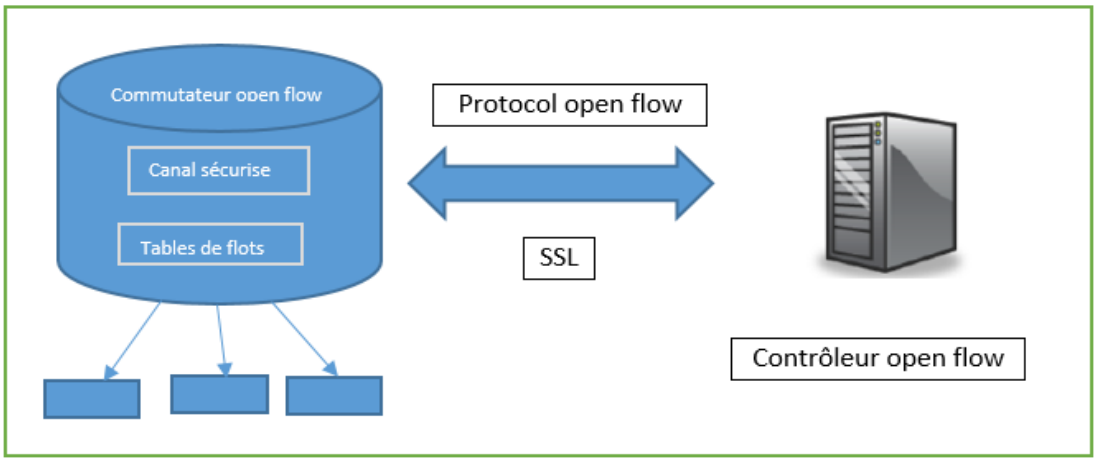


Figure 3: Switch SDN matériel

6 .Le mode opérationnel de contrôleur

Tableaux de flux. À l'aide du protocole de commutation Open Flow, le contrôleur peut ajouter, mettre à jour et supprimer des entrées de flux dans les tables de flux, à la fois de manière réactive (en réponse aux paquets) et de manière proactive.

Des entrées de flux réactives sont créées lorsque le contrôleur apprend dynamiquement où se trouvent les périphériques dans la topologie et doit mettre à jour les tables de flux sur ces périphériques pour établir une connectivité de bout en bout. Par exemple, étant donné que les commutateurs dans un environnement Open Flow pur sont simplement des transitaires de trafic, toute logique rationnelle doit d'abord être dictée et programmée par le contrôleur. Ainsi, si un hôte sur le commutateur A besoin de parler à un commutateur hôte B, des messages seront envoyés au contrôleur pour savoir comment accéder à cet hôte. Le contrôleur apprendra les tables d'adresses MAC hôtes des commutateurs et comment ils se connectent, en programmant la logique dans les tables de flux de chaque commutateur. Il s'agit d'une entrée de flux réactif.

Les entrées de flux proactives sont programmées avant l'arrivée du trafic. S'il est déjà connu que deux appareils doivent ou ne doivent pas communiquer, le contrôleur peut programmer ces entrées de flux sur les points de terminaison Open Flow à l'avance.

7 OPEN FLOW

Open Flow a débuté à l'Université de Stanford en 2008 [19]. Le but du projet était de donner aux chercheurs un outil pour mettre en œuvre leurs protocoles expérimentaux dans les réseaux.
La spécification est contrôlée et définie par la fondation de réseau ouvert à but non lucratif (ONF), créée en 2011 par un groupe d'éditeurs [20] et dirigée par un conseil d'administration de sept sociétés qui possèdent et exploitent certains des plus grands réseaux au monde (Deutsche Telekom, Facebook, Google, Microsoft, Verizon, Yahoo et NTT). La plupart des fournisseurs de matériel réseau tels que HP, IBM et CISCO proposent des commutateurs et des routeurs qui utilisent le protocole Open Flow [21].
Les commutateurs compatibles Open Flow sont de deux types principaux :

- ***Open Flow uniquement*** : Les commutateurs ne prennent en charge que les opérations Open Flow, c'est-à-dire que tous les paquets sont traités par le pipeline Open Flow.
- ***Open Flow hybride*** : Les commutateurs hybrides Open Flow prennent en charge à la fois les opérations Open Flow et les opérations de commutation Ethernet normales, donc la commutation et le routage L2 et L3 traditionnels.

Le réseau Open Flow se compose de trois composants principaux : un contrôleur, un commutateur Open Flow et le protocole Open Flow.

7.1 Contrôleurs SDN

Le contrôleur est l'élément principal d'un réseau SDN, il est considéré comme son système d'exploitation. Le Controller centralise les décisions pour toute la communication qui passe par les appareils et donne un aperçu du réseau. Une description générale d'un contrôleur SDN serait système logiciel ou collection de systèmes qui Ensemble fournissent [22] Une gestion de l'état du réseau Un modèle de données de haut niveau qui capture la relation entre les ressources gérées les politiques et d'autres services fournis par le contrôleur. Une interface de programmation d'application qu'exposent les services offerts par le contrôleur pour les applications.

7.2 L'Open Flow switch

Le commutateur Open Flow possède une ou plusieurs tables de flux. Une table de flux est un ensemble d'entrées de flux. Une entrée de flux est utilisée pour faire correspondre et traiter les paquets. Il se compose de nombreux champs pour faire correspondre les paquets, et un ensemble de rencontres pour suivre les paquets et instructions à appliquer [23]. Le commutateur

Open Flow utilise un canal Open Flow pour communiquer avec le contrôleur Open Flow. Un commutateur Open Flow comporte les deux composants suivants :

7.2.1 Flow & Group Tables

Chaque entrée de la table d'écoulement se compose de trois champs, d'un en-tête de correspondance, de compteurs et d'un ensemble d'instructions à appliquer aux paquets correspondants [24].

- L'en-tête du paquet est spécifique à l'écoulement et le définit. Cet en-tête est presque un dix-Tuple. Ses champs contiennent des informations telles que l'ID VLAN, les ports de source et de destination, l'adresse IP et la source et la destination Ethernet.
- L'action spécifie comment les paquets d'un flux seront traités. Une action peut être l'un des éléments suivants :

  -Transférer le paquet vers un port ou des ports donnés.

  - Déposez le paquet.

  -Transférer le paquet au contrôleur.
- Les statistiques incluent des informations telles que le nombre de paquets, le nombre d'octets, le temps depuis que le dernier paquet correspondait au flux, etc. pour chaque type de flux. La plupart du temps, les compteurs sont utilisés pour garder une trace du nombre de paquets et d'octets pour chaque flux et le temps écoulé depuis l'initiation du flux.

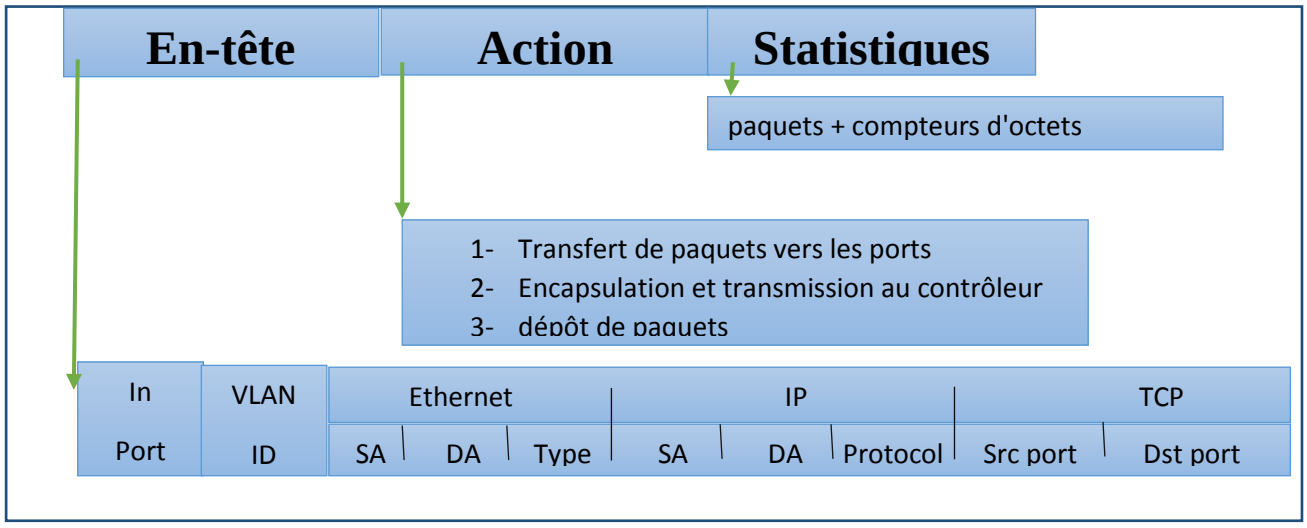


Figure 4: Flow & Group Tables

7.2.2 The Open Flow Channel

C'est un canal sécurisé entre le commutateur open Flow et le contrôleur open Flow. Il permet la communication en permettant au plan de contrôle d'envoyer des instructions, de recevoir des demandes ou d'échanger des informations. Tous les messages sont cryptés, en utilisant la sécurité de la couche de transport (TLS). Les messages commencent par l'en-tête open Flow. Cet en-tête spécifie la version du protocole open Flow, le type de message, la longueur du message et l'ID de transaction du message [23].

Le canal open Flow a trois types de messages :

7.2.2.1 Controller-to Switch

Ces messages sont lancés par le contrôleur pour gérer ou inspecter l'état d'un commutateur et peuvent ou non nécessiter une réponse du commutateur.

***Features*:** le contrôleur peut demander l'identité ET les capacités d'une Switch ; cela se passe en envoyant une requête.

Modify-state : ce type de message permet de gérer l'état des Switch ; ils permettent de modifier ou ajouter, effacer les entrées dans les tables open flow.

Configuration **:** le contrôleur peut définir les paramètres de configuration des Switch. Ce dernier répond aux requêtes envoyées par le contrôleur.

Read-state **:** le contrôleur utilise ces messages pour collecter différentes informations des Switch, tel que les statistiques, les capacités et sa configuration actuelle.

Packet-out : ces messages sont utilisés pour le transfert de paquets reçus par les messages Packet-in. Le message doit contenir un paquet entier ou l'ID du buffer faisant référence à un paquet stocké dans les Switch. Si la liste d'actions du message est vide, le paquet sera détruit.

Barrier : utilisé par le contrôleur pour recevoir des notifications de l'opération terminée.

7.2.2.2 Symétrique

Ce sont des messages envoyés sans demande préalable, dans les deux sens. Bonjour, les messages sont généralement envoyés entre le contrôleur et les commutateurs lorsque la connexion est établie pour la première fois. Les messages de demande d'écho et de réponse peuvent être utilisés par les commutateurs ou le contrôleur pour mesurer la latence ou la bande passante d'une connexion de commutateur contrôleur.

Hello : envoyer pour vérifier la connectivité entre le contrôleur et les Switch.

Echo: une fois la connexion établie, ces messages sont échangés entre le contrôleur ET Le switch. Chaque message ECHO_REQUEST doit être acquitté par un ECHO_REPLY.

Error: utilisé par le contrôleur et les Switch pour signaler un problème de connexion.

7.2.2.3 Asynchrone

Ce sont des messages envoyés au contrôleur sans avoir été demandés précédemment, car ils communiquent l'arrivée des paquets, des modifications des états ou des erreurs. Un exemple typique est le message de paquets, qui peut être utilisé par un commutateur pour envoyer un paquet au contrôleur lorsqu'il n'y a pas de correspondance de table de flux.

- ***Packet-in*** : avec ce message, le switch transfert le contrôle du paquet au contrôleur.
- ***Flow-removed***: le switch informe le contrôleur de la suppression d'une entrée dans la Table des flux.
- ***Port-satus*** : avec ce message, le switch informe le contrôleur d'un changement sur un Port.
- ***Role-status*** **:** quand un nouveau contrôleur est aux commandes, le switch envoie un Rôle-status à son contrôleur.

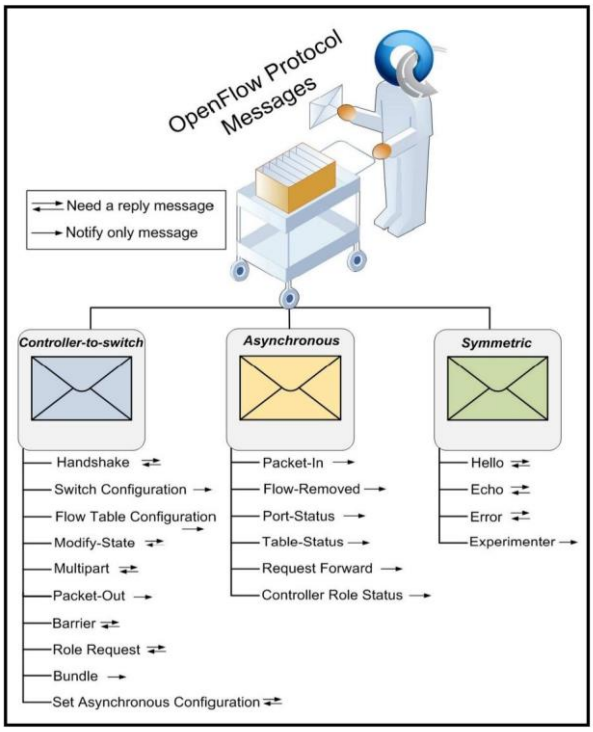


Figure 5: The Open Flow Protocol Messages

7.3 The Open Flow protocole

Open Flow est le protocole utilisé pour gérer l'interface sud de l'architecture SDN généralisée. Il s'agit de la première interface standard définie pour faciliter l'interaction entre les plans de contrôle et de données de l'architecture SDN. Open Flow fournit un accès logiciel aux tables de flux qui indiquent aux commutateurs et aux routeurs comment diriger le trafic réseau.

À l'aide de ces tableaux de flux, les administrateurs peuvent modifier rapidement la disposition du réseau et le flux de trafic. De plus, le protocole Open Flow fournit un ensemble de base d'outils de gestion qui peuvent être utilisés pour contrôler des fonctionnalités telles que les changements de topologie et le filtrage de paquets.

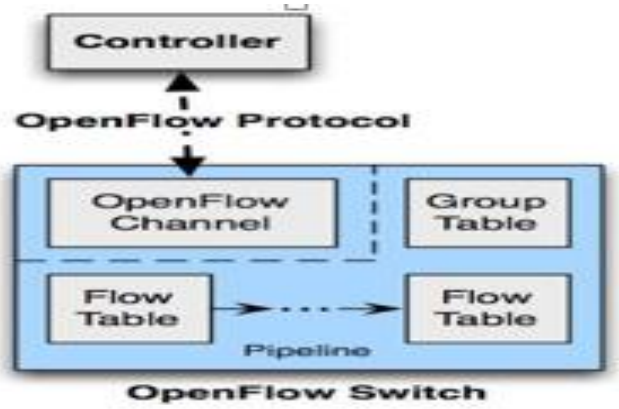


Figure 6: The Open Flow protocol

8 15Spécifications Open flow

Nous résumons les spécifications d'Open flow dans le tableau suivant :

Open Flow version	Date	Features
Open Flow 1.0.0 [25]	Dec,2009	Single flow table, IPv4
Open Flow 1.1.0 [26]	Feb,2011	Multiple flow table, group table, MPLS support and VLAN
Open Flow 1.2.0	Dec,2011	IPv6, multiple controllers
Open Flow 1.3.0 [27]	Jun,2012	Single flow measure, IPv6 extend header, Meters for QoS Capabilities
Open Flow 1.4.0 [28]	Oct,2013	Flow table synchronization mechanism, bundling message
Open Flow 1.5.0 [29]	Dec,2014	Data packet type identification process, egress table, scheduled bundle expansion

Tableau 2 : Spécifications d'Open flow

9 Les avantages de SDN

- Facilité de la conception et du contrôle du réseau, et surtout l'existence d'une structure de contrôle centralisée du trafic dans le réseau.
- La possibilité d'exécution d'une ou plusieurs règles de réseaux sur un équipement standard.
- L'orchestration et l'automatisation des réseaux de différents fournisseurs permet de provisionner des services réseaux, rapidement et à grande échelle, en réduisant le risque de l'erreur humain.
- La vue globale des réseaux par le contrôleur permet de remplacer les protocoles de routage distribués (OSPF, EIGRP, RIP…) par des mécanismes plus complexes.
- Faible coût à comparer avec les équipements réseaux actuels que nous achetons fermés qu'il est impossible de développer ou de fusionner avec d'autres produits, ainsi que de réduire le nombre d'ingénieurs, et gagner le temps de travail de plus de 50 %.
- L'acheminement de services dynamiques virtuels se définit par une politique des flux selon les exigences des entreprises.
- Ne pas être dépendant d'un seul fournisseur de produit (comme Cisco ou Juniper), ce qui permet l'existence d'une flexibilité dans le réseau, cela permettra à l'administrateur de faire des choix indépendants des meilleurs éléments au sein du réseau.
- L'existence des SDN va augmenter le taux d'innovation au niveau de l'infrastructure réseau, et cela conduira à l'apparition de nouvelles idées, par exemple, les développeurs peuvent tester des applications au sein du réseau sans affecter les performances du réseau ou sur d'autres services.

10 Inconvénients des réseaux SDN

- Dépendance vis-à-vis du contrôleur : Dans les réseaux SDN, la défaillance ou la surcharge du contrôleur central peut entraîner des problèmes de connectivité dans l'ensemble du réseau. La fiabilité et la disponibilité du contrôleur sont donc critiques.
- Complexité initiale : L'adoption des réseaux SDN peut être complexe au départ, car elle nécessite des compétences spécifiques en programmation et en gestion de réseau. Les entreprises peuvent avoir besoin de former leur personnel ou d'engager des experts pour mettre en place et maintenir le réseau SDN.
- Interopérabilité : L'interopérabilité entre les différents équipements réseau et les différents fournisseurs peut être un défi dans les réseaux SDN. Les normes et les protocoles ne sont pas toujours uniformément pris en charge, ce qui peut limiter le choix des équipements compatibles.

11 Conclusion

Le Software Defined Networking annonce des changements importants sur les réseaux dans les années à venir, c'est une technologie émergente qui est susceptible de révolutionner les activités de réseau traditionnelles. Ceux-ci vont voir leur architecture profondément évoluer, facilitant des nouveaux usages, permettant aux administrateurs réseau de gérer et de contrôler automatiquement et dynamiquement un grand nombre de dispositifs, de services, de topologie, de trajets de trafic et de gestion de paquets (qualité de service) en utilisant un langage et des API de haut niveau.

Chapitre 2: réseaux traditionnels

1 Introduction

Le monde de l'informatique et des technologies de l'information a connu une évolution sans précédent ces dernières décennies. Les réseaux informatiques ont joué un rôle essentiel dans cette évolution en permettant la communication et le partage de données à travers le monde. Au fil des années, différents types de réseaux ont vu le jour, chacun ayant ses avantages et ses limites. Parmi ces réseaux, les réseaux traditionnels ont été les premiers à être développés et ont largement contribué à la mise en place des infrastructures de communication que nous utilisons aujourd'hui. Dans ce chapitre, nous allons examiner en détail les réseaux traditionnels, leur fonctionnement, leur architecture et leur place dans l'évolution des technologies de l'information.

2 Définition du routage

Le routage est un processus qui consiste à sélectionner les chemins appropriés dans un réseau pour acheminer les données (paquets) d'un émetteur à un ou plusieurs destinataires, en utilisant des adresses IP. Dans les premiers réseaux, les tables de routage étaient statiques et où les chemins sont prédéfinis et doivent être configurés manuellement .Toutefois, avec l'expansion et la complexité croissante des réseaux, des protocoles de routage dynamique ont été développés pour échanger des informations de routage avec d'autres systèmes du réseau et mettre à jour automatiquement les tables de routage.

3 Principe de fonctionnement du routage

Lorsqu'un ordinateur envoie un message à un autre situé en dehors de son réseau, ce message est transmis à un routeur. Le routeur effectue alors les opérations suivantes :

- Lecture de l'adresse du destinataire.
- Consultation de sa table de routage pour déterminer le chemin à suivre pour atteindre cette destination.
- Transmission du message au routeur suivant (ou directement au destinataire s'il est à proximité).

4 Architecture des réseaux traditionnels

L'architecture des réseaux traditionnels se compose de quatre couches ou plans de logiciels. À l'intérieur de chaque périphérique réseau et de sécurité, à savoir chaque commutateur, routeur et pare-feu :

- Le plan de transfert achemine les paquets réseau aussi rapidement que possible.
- Le plan de contrôle décide de la direction des flux réseau et assure la fluidité du trafic en décodant les protocoles. Il est intimement relié au matériel.

- Le plan de services permet d'exploiter et de déployer des services plus complexes sur l'appareil en question, par exemple un pare-feu.
- Le plan de gestion fournit les instructions de base indiquant comment le périphérique réseau doit interagir avec le reste du réseau, et est accessible via une interface de ligne de commande (CLI) pour la configuration du périphérique. [30]

La figure 7 ci-dessous illustre l'architecture et le fonctionnement des équipements réseau traditionnels, qui présentent des inconvénients tels que :

Complexité : l'ajout ou la modification d'équipements ainsi que la mise en place de politiques réseaux sont des tâches complexes, chronophages et susceptibles de causer des interruptions de service. Cette situation décourage les modifications et l'évolution du réseau.

Difficulté à s'adapter à l'échelle : l'incapacité à disposer d'un réseau capable de s'adapter au trafic oblige les opérateurs à surdimensionner leurs réseaux, ce qui complexifie la gestion du plan de contrôle.

Dépendance aux fabricants : les fabricants proposent des produits avec des durées de vie limitées et des interfaces propriétaires ou non standardisées, ce qui limite la capacité des opérateurs réseau à adapter le réseau à leurs besoins [31]

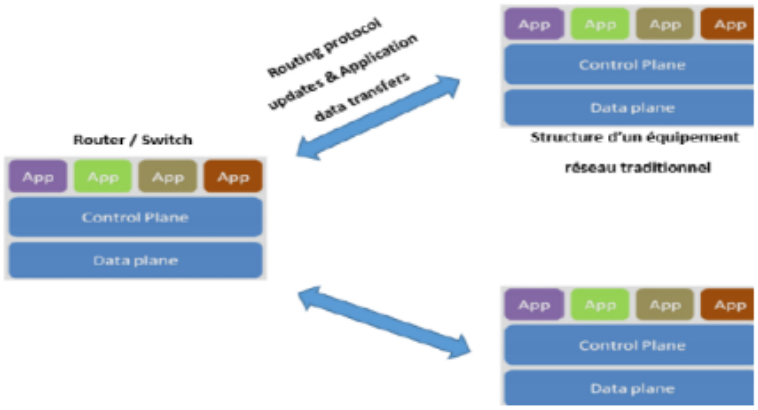


Figure 7: Fonctionnement des structures des équipements réseau traditionnel

5 Méthodes de routage

5.1 Routage Direct

Le routage direct permet de transmettre des datagrammes entre deux machines qui sont connectées sur le même réseau physique, sans avoir besoin de passer par un routeur. Les datagrammes sont encapsulés dans des trames correspondant au type du réseau local et sont

directement acheminés vers la machine de destination. Ce type de routage est le plus simple et le plus rapide car il ne nécessite pas d'analyse de la destination ou de recherche dans une table de routage.

5.2 Routage indirect

Le routage indirect est utilisé lorsque la destination se trouve sur un sous-réseau différent mais sur le même réseau physique que l'émetteur. Dans ce cas, un routeur est nécessaire pour acheminer le trafic entre les deux sous-réseaux. Le datagramme IP est envoyé au routeur, qui détermine le chemin optimal vers la destination finale en se basant sur les informations contenues dans les tables de routage. Le routage indirect peut également se faire par routage par défaut, où le routeur par défaut est utilisé si aucune entrée n'est trouvée dans la table de routage. [32]

6 Table de routage

La table de routage est une structure de données utilisée par les routeurs pour déterminer la meilleure route pour acheminer les paquets de données vers leur destination. Elle contient des informations sur les réseaux connectés directement au routeur ainsi que les réseaux distants accessibles par le biais d'autres routeurs.

Chaque entrée dans la table de routage correspond à une destination spécifique et indique la prochaine étape pour atteindre cette destination. Les éléments clés de chaque entrée de la table de routage incluent :

- ***Méthode de routage*** : le protocole de routage qui a appris cette route, par exemple OSPF, BGP, RIP, etc.
- ***Réseau et masque*** : l'adresse IP du réseau de destination et le masque de sous-réseau correspondant.
- ***Distance administrative*** : la préférence d'une route par rapport à une autre, utilisée pour résoudre les conflits de routes. Chaque protocole de routage a une valeur par défaut pour la distance administrative.
- ***Valeur de métrique*** : la mesure de la qualité de la route, utilisée pour sélectionner la meilleure route parmi toutes celles apprises par un protocole de routage.
- ***Prochaine passerelle (Gateway)*** : l'adresse IP de la passerelle ou du prochain routeur sur le chemin vers la destination.
- ***Interface de sortie*** : l'interface du routeur par laquelle les paquets de données seront envoyés pour atteindre la destination.

- En fonction des paramètres de routage, la table de routage peut être remplie de manière statique ou dynamique. Dans un environnement de routage dynamique, les routeurs échangent des informations de routage pour mettre à jour automatiquement leur table de routage.

6.1 Routage par défaut

Si aucun itinéraire correspondant n'est trouvé dans la table de routage IP pour la destination spécifiée dans le datagramme, il sera alors envoyé à une passerelle spéciale appelée passerelle par défaut (Default Gateway), dont l'adresse est généralement enregistrée dans la table de routage.

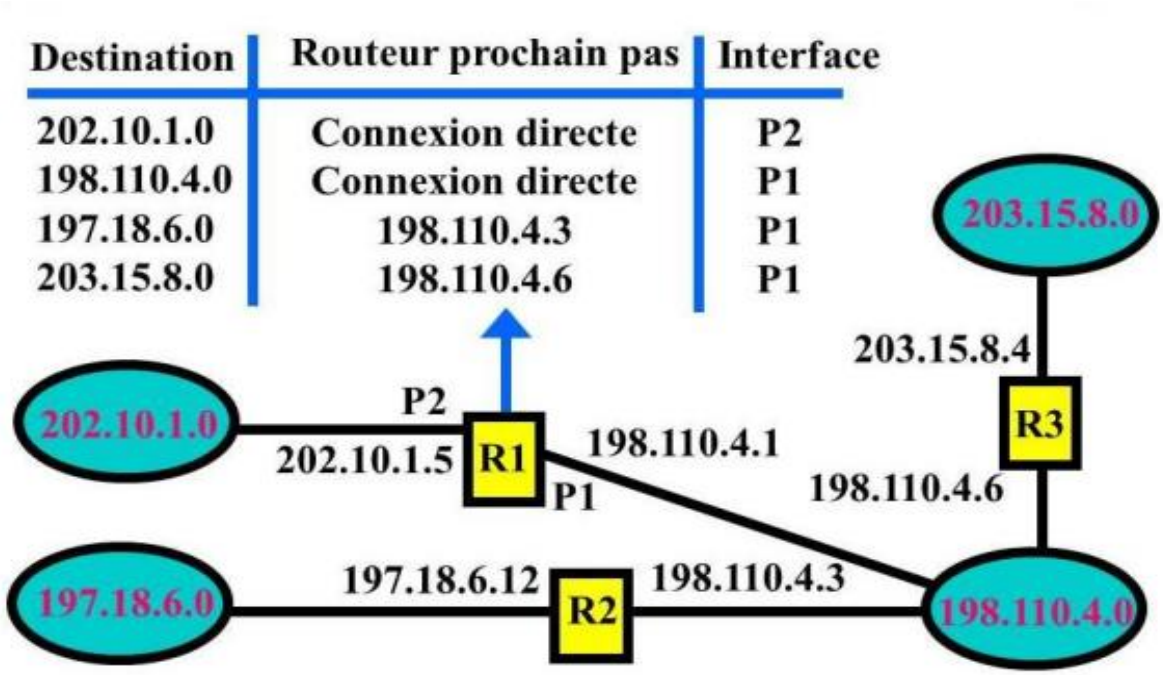


Figure 8: Exemple d'une table de routage

7 Les différents types de routage

7.1 Le routage statique

Dans le cas du routage statique, l'administrateur réseau remplit manuellement les tables de routage. Cette méthode est généralement utilisée pour les petits réseaux ou les réseaux de périphérie. L'administrateur doit gérer manuellement les itinéraires pour chaque unité de routage du réseau, et les itinéraires statiques ne s'adaptent pas aux modifications de l'environnement du réseau. Les informations de routage doivent donc être mises à jour manuellement chaque fois qu'une modification topologique est apportée au réseau.

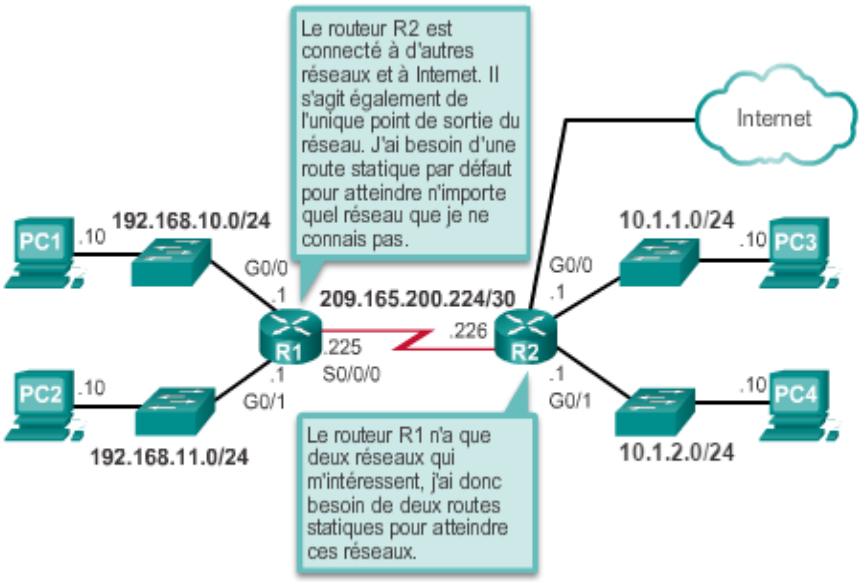


Figure 9: Scenarios de routes statique et par défaut

7.1.1 Avantages du routage statique

- Le traitement processeur est minimal, car les tables de routage sont préconfigurées et ne nécessitent pas de calculs de mise à jour en temps réel.
- Il y a moins de surcharge de trafic réseau par rapport au routage dynamique.

7.1.2 Inconvénients du routage statique

- La configuration et la maintenance sont chronophages, car l'administrateur doit mettre à jour manuellement les tables de routage pour chaque unité de routage du réseau.
- La configuration manuelle peut conduire à des erreurs, en particulier dans les grands réseaux.
- L'administrateur doit intervenir régulièrement pour maintenir les informations de routage à jour en cas de changements topologiques.
- Le routage statique ne convient pas bien aux réseaux en expansion, car la maintenance peut devenir fastidieuse à mesure que le réseau se développe.
- La mise en place d'un routage statique exige une connaissance approfondie de l'ensemble du réseau pour une implémentation correcte.

7.2 Le Routage dynamique

Le routage dynamique permet de remplir automatiquement les tables de routage en utilisant un protocole configuré pour établir la topologie du réseau. Les tables de routage sont ainsi

constamment mises à jour pour prendre en compte les changements dans l'environnement du réseau, comme la rupture d'un lien sur un routeur. Ce type de routage permet également de sélectionner automatiquement la meilleure route disponible pour atteindre une destination donnée à travers le réseau.

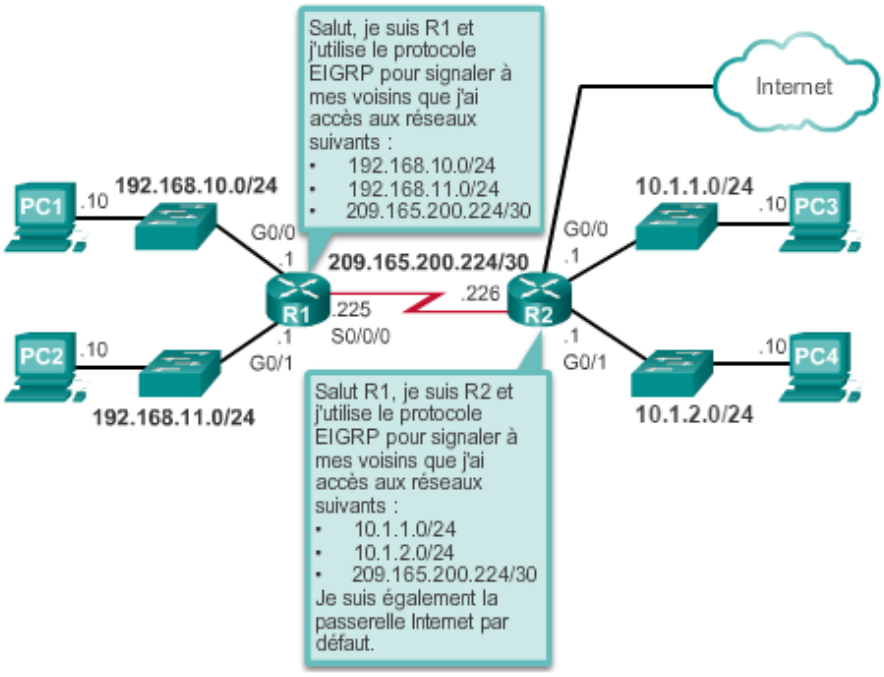


Figure 10:Scenarios de routage dynamique

7.2.1 Avantages du routage dynamique

- Réduction des tâches de maintenance pour l'administrateur lors de l'ajout ou de la suppression de réseaux, car les protocoles de routage dynamique se chargent de la mise à jour des tables de routage.
- Les protocoles de routage dynamique réagissent automatiquement aux changements topologiques du réseau, ce qui assure une grande fiabilité de la communication.
- La configuration est moins sujette aux erreurs, car les protocoles de routage dynamique ne nécessitent pas de configuration manuelle des tables de routage.
- Le routage dynamique est plus évolutif et s'adapte bien aux réseaux en expansion.

7.2.2 Inconvénients du routage dynamique

- Le routage dynamique utilise davantage de ressources du routeur, notamment le cycle de processeur, la mémoire et la bande passante de liaison.
- Les administrateurs doivent posséder des connaissances plus approfondies pour la configuration, la vérification et le dépannage des protocoles de routage dynamique.

7.3 Routage hybride

Le routage hybride est un type de routage qui combine les avantages du routage statique et du routage dynamique.

8 Les protocoles de routage

Il y a deux grandes catégories de protocoles de routage pour les réseaux IP : les protocoles de routage interne IGP (Interior Gateway Protocol) et les protocoles de routage externe EGP (Exterior Gateway Protocol). Le réseau Internet est divisé en systèmes autonomes (AS) ou zones de responsabilité, qui sont des groupes de routeurs et de réseaux connectés administrés par une même organisation. Dans un système autonome, un protocole de routage interne de type IGP est utilisé pour échanger des informations de routage entre les différents routeurs. Pour l'échange de routage entre systèmes autonomes différents, un protocole de routage externe de type EGP est utilisé.

Il peut arriver qu'un protocole de routage découvre plusieurs chemins menant à la même destination. Pour sélectionner le meilleur chemin, le protocole doit être capable d'évaluer et de différencier les chemins disponibles. Une métrique est utilisée à cette fin. La métrique est une valeur utilisée par les protocoles de routage pour affecter des coûts d'accès aux réseaux distants. Elle est utilisée pour déterminer le meilleur chemin en présence de plusieurs chemins vers le même réseau distant. Selon le protocole de routage utilisé, différentes métriques peuvent être prises en compte lors de la décision de routage.

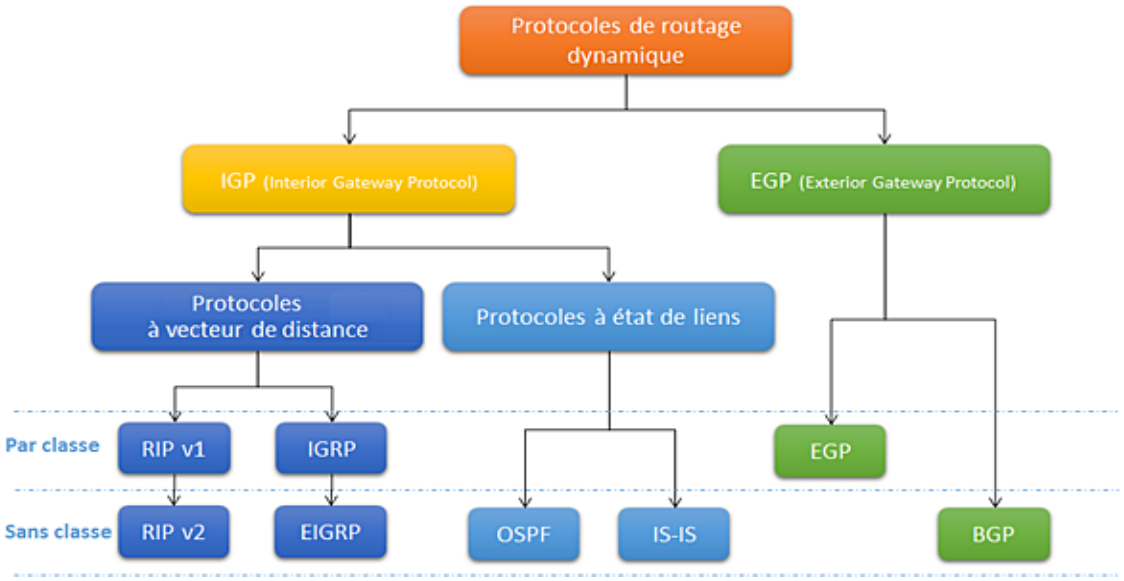


Figure 11:Classification des protocoles de routage

9 Les protocoles de routage interne

9.1 Les protocoles à vecteur de distance

Les protocoles de routage à vecteur de distance transmettent régulièrement des copies de leur table de routage d'un routeur à l'autre. Chaque routeur reçoit la table de routage de son voisin immédiat et utilise ces informations pour mettre à jour sa propre base de données contenant les informations sur la topologie du réseau. Ce processus est répété étape par étape dans toutes les directions entre les routeurs immédiatement voisins, permettant ainsi au protocole de cumuler les distances réseau. Cependant, ces protocoles ne permettent pas à un routeur de connaître la topologie exacte du réseau. [33]

Les protocoles à vecteur de distance sont de nature distribuée, itérative et asynchrone. Ils sont distribués car les calculs se font au niveau de chaque nœud, à partir des informations fournies par les voisins immédiats, et les résultats sont partagés de la même manière. Ils sont itératifs car le processus de mise à jour de la table de routage se répète jusqu'à ce qu'il n'y ait plus d'informations à échanger entre les nœuds voisins, et s'arrête de lui-même. Enfin, ils sont asynchrones car ils n'imposent pas à tous les routeurs de travailler ensemble. [34]

Les protocoles de routage dynamique à vecteur de distance incluent RIP, IGRP et EIGRP.

9.1.1 Le protocole RIP

Le protocole RIP, initialement défini dans le document RFC 1058, possède les caractéristiques suivantes :

- Il utilise le nombre de sauts comme mesure de sélection de chemin.
- Si le nombre de sauts pour un réseau est supérieur à 15, le protocole RIP ne peut pas fournir de route à ce réseau.
- Par défaut, les mises à jour de routage sont diffusées ou multi diffusées toutes les 30 secondes.

  Par exemple, dans la figure présentée, R1 sait que la distance pour atteindre le réseau 172.16.3.0/24 est de 1 saut et que la direction est celle de l'interface S0/0/0 vers R2.

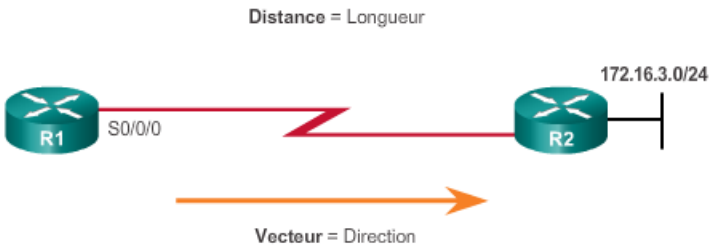


Figure 12: Les protocoles à vecteur de distance

Pour R1 le réseau 172.16.3.0/24 est éloigné d'un saut (distance) il peut être atteint via R2 (Vecteur)

Il existe actuellement deux versions à ce jour, RIPv1 ainsi que RIPv2

RIPv1 RIPv1 est défini dans la RFC 1058. Cette version ne prend pas en charge le masque des sous-réseaux de longueur variable ni de l'authentification des routeurs. Les routes sont envoyées en broadcaste.

RIPv2 RIPv2 est défini dans la RFC 2453. Cette version développée en 1993, a été conçue pour permettre au protocole de répondre aux contraintes des réseaux actuels (découpages des réseaux IP en sous-réseaux, authentification par mot de passe). Avec cette version, les routes sont envoyées à l'adresse multicast 224.0.0.9.

9.1.2 Le protocole IGRP et EIGRP

Le protocole IGRP est un protocole de routage intérieur à vecteur de distance utilisé par les routeurs pour échanger des informations de routage au sein d'un système autonome. Contrairement au protocole RIP, il n'a pas de limite de taille de réseau, mais ne supporte pas les masques de sous-réseau variables et est actualisé toutes les 90 secondes. Il offre plus de critères de sélection de chemin, tels que la bande passante, le délai et la charge réseau, avec la possibilité de donner manuellement une priorité à chacun de ces critères.

Le protocole EIGRP (Enhanced Interior Gateway Routing Protocol) est une version améliorée du protocole IGRP, développé par Cisco pour pallier les limitations associées au routage dans de grands réseaux. Il utilise l'algorithme DUAL (Diffusing Update Algorithm) développé par SRI International, qui permet une convergence plus rapide et efficace du réseau. EIGRP prend également en charge les masques de sous-réseau variables et l'authentification des routeurs.

Caractéristiques de l'IGRP

Le protocole IGRP utilise une métrique basée sur différents critères tels que la bande passante, la charge, la fiabilité et la longueur du chemin pour déterminer le chemin optimal entre deux réseaux. IL est conçu pour avoir une convergence rapide en cas de changement de topologie, ce qui permet aux routeurs de récupérer rapidement après une panne de lien. De plus, IGRP prend en charge la hiérarchie de routage, ce qui permet de diviser le réseau en groupes logiques pour réduire la taille de la table de routage et améliorer les performances du routage. Ce protocole est également adapté aux entreprises de taille moyenne grâce à sa capacité à gérer des réseaux de grande taille. Enfin, IGRP prend en charge l'authentification, ce qui permet de garantir que seuls les routeurs autorisés peuvent participer à la communication du protocole IGRP.

Caractéristiques de l'EIGRP

- ***Routage basé sur la distance*** : EIGRP utilise une métrique basée sur la bande passante, la charge, la fiabilité et la longueur du chemin pour déterminer le meilleur chemin entre deux réseaux.
- ***Convergence rapide*** : Ce protocole est conçu pour avoir une convergence rapide en cas de changement de topologie, ce qui signifie que les routeurs peuvent récupérer rapidement après une panne de lien.
- ***Hiérarchie de routage*** : Il prend en charge la hiérarchie de routage, ce qui permet de diviser le réseau en groupes logiques pour réduire la taille de la table de routage et améliorer les performances du routage.
- ***Protocole hybride*** : EIGRP est un protocole de routage hybride qui combine les avantages des protocoles de routage à vecteur de distance et à état de lien. Cela signifie qu'il utilise à la fois des informations de voisinage (vecteur de distance) et des informations de topologie (état de lien) pour prendre des décisions de routage.
- ***Scalabilité*** : Il est capable de gérer des réseaux de grande taille, ce qui le rend adapté aux entreprises de taille moyenne.
- ***Sécurité*** : Il prend en charge l'authentification, ce qui permet de garantir que seuls les routeurs autorisés peuvent participer à la communication du protocole.

10 Protocoles d'état de lien

Les protocoles de cette famille impliquent une communication bidirectionnelle entre chaque routeur du réseau, permettant à chacun de construire une vue complète de la topologie du réseau ainsi qu'une table de routage prenant en compte les meilleures routes disponibles. Les décisions

de routage sont basées sur des métriques telles que la qualité du lien, l'encombrement, le type de flux à transmettre, les restrictions de qualité de service et le coût financier. Cette approche permet une construction plus rapide des tables de routage que le routage à vecteur de distance. Tous les paquets sont alors acheminés sur le chemin optimal déterminé par le protocole de routage.

10.1 OSPF (Open Shortest Path First)

Dans le protocole OSPF, les routeurs établissent des liens directs avec leurs voisins et communiquent les réseaux auxquels ils sont connectés en utilisant des messages "hello" envoyés régulièrement. Ces messages "LinkState Advertisement" (LSA) sont ensuite propagés à tous les routeurs du réseau, créant ainsi une base de données d'état des liens (LSDB) qui est identique pour tous les routeurs dans la même zone. À l'aide de l'algorithme de Dijkstra "Shortest Path First" (SPF), chaque routeur détermine la route la plus rapide vers chaque réseau connu dans la LSDB. Le protocole OSPF ne permet pas de filtrer ou de résumer les routes dans une zone car la cohérence dans le calcul SPF est essentielle pour le bon fonctionnement du protocole. En cas de changement de topologie, de nouveaux LSA sont propagés et l'algorithme SPF est réexécuté sur chaque routeur pour mettre à jour les tables de routage.

Caractéristiques du protocole OSPF

- ***Routage à état de lien*** : OSPF utilise un algorithme de routage, ce qui signifie que les routeurs échangent des informations sur l'état de leurs liens avec les autres routeurs du réseau pour déterminer les meilleurs chemins possibles.
- ***Routage basé sur le coût*** : Il utilise un coût pour déterminer le chemin le plus court entre deux réseaux. Le coût est déterminé par la bande passante disponible sur chaque lien.
- ***Scalabilité*** : OSPF est conçu pour être hautement évolutif, ce qui signifie qu'il peut gérer de grands réseaux sans impacter les performances.
- ***Hiérarchie*** : Il utilise une hiérarchie de routage, où les routeurs sont organisés en domaines et en zones. Cela permet de réduire la taille de la table de routage et d'améliorer les performances du routage.
- ***Convergence rapide*** : Ce protocole est conçu pour avoir une convergence rapide, ce qui signifie qu'il peut rétablir la connectivité du réseau rapidement en cas de panne de lien.

- ***Sécurité :*** Il prend en charge l'authentification pour garantir que seuls les routeurs autorisés peuvent participer à la communication du protocole.

11 Les protocoles de routage externes

La principale information à retenir sur les protocoles externes est que la plupart des systèmes ne les utilisent jamais. Ces protocoles permettent l'échange d'informations d'acheminement entre des systèmes autonomes, appelées informations d'accessibilité, qui fournissent des informations sur les réseaux accessibles via un système autonome spécifique. Le premier protocole de routage inter domaine, EGP, avait des problèmes mais répondait aux besoins de l'époque et a été utilisé pendant de nombreuses années. Il a été remplacé par BGP, qui transmet la suite des numéros de systèmes autonomes rencontrés sur le chemin vers l'adresse IP de destination, permettant à chaque routeur BGP de calculer et d'annoncer sa route préférée à ses voisins. Les politiques complexes peuvent être implémentées avec BGP en utilisant des informations de configuration manuelle pour classer les routes par ordre de préférence, car BGP ne transmet pas le coût vers une destination. Pour déterminer la route préférée parmi plusieurs options, chaque routeur BGP doit construire une fonction locale qui prend en entrée les informations de mise à jour BGP concernant une destination spécifique et qui renvoie un nombre. Ensuite, les différentes routes [35]

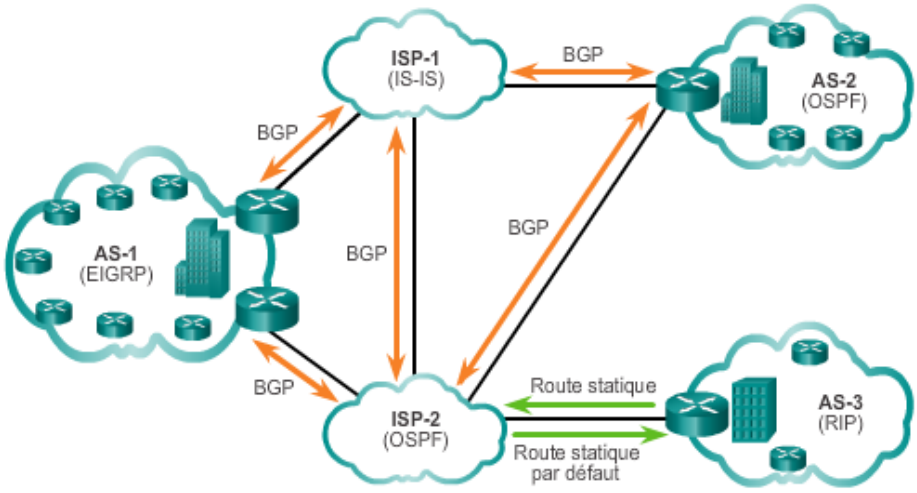


Figure 13:Les protocoles de routage externes

Caractéristiques de BGP

BGP (Border Gateway Protocol) est un protocole de routage utilisé pour échanger des informations de routage entre les systèmes autonomes (AS) dans un réseau Internet. Ses principales caractéristiques sont les suivantes :

- Protocole de routage basé sur des chemins, il sélectionne le chemin le plus court pour atteindre une destination donnée en utilisant des critères tels que la préférence de l'opérateur de réseau, le coût, la bande passante, etc.
- Utilise un système de politiques de routage qui permet aux opérateurs de réseau de contrôler et de gérer la façon dont les routes sont annoncées et propagées dans le réseau. Les politiques peuvent être utilisées pour bloquer certaines routes, préférer certaines routes, etc.

- Protocole de routage externe, utilisé pour échanger des informations de routage entre différents systèmes autonomes. Conçu pour gérer des réseaux de grande taille et complexes.

- Fiable et stable, conçu pour minimiser les erreurs de routage et les boucles de routage. Utilise un système de vérification de cohérence de routage pour s'assurer que les routes annoncées sont valides et qu'il n'y a pas de boucles de routage.

- Évolutif, conçu pour gérer des milliers de routes différentes et pour s'adapter aux changements dans le réseau. Largement utilisé dans les réseaux Internet de grande taille et de haute complexité.

12 Avantages des réseaux traditionnels

- ***Maturité*** : Les réseaux traditionnels sont une approche établie et bien comprise dans l'industrie des réseaux. De nombreuses entreprises ont déjà déployé des infrastructures de réseau traditionnelles et ont une expertise dans leur gestion.
- ***Performance*** : Les réseaux traditionnels peuvent offrir une latence plus faible et des performances réseau optimisées dans des environnements spécifiques, notamment pour les applications temps réel ou hautement sensibles à la latence.
- ***Indépendance vis-à-vis des tiers*** : Avec les réseaux traditionnels, les entreprises ont un contrôle direct sur leurs équipements réseau et ne dépendent pas d'un tiers pour la gestion et le contrôle du réseau.

13 Inconvénients des réseaux traditionnels

- ***Complexité de gestion*** : Les réseaux traditionnels peuvent être complexes à gérer, surtout lorsqu'ils sont vastes et comprennent de nombreux équipements réseau. Les configurations manuelles et les mises à jour peuvent être fastidieuses et propices aux erreurs humaines.
- ***Manque de flexibilité*** : Les réseaux traditionnels peuvent être moins flexibles en termes de changements de configuration et de déploiement rapide de nouvelles fonctionnalités.

Des modifications peuvent nécessiter des interventions manuelles sur chaque équipement réseau, ce qui peut être long et coûteux.

- ***Difficulté d'adaptation*** : Les réseaux traditionnels peuvent être moins adaptés aux exigences évolutives et dynamiques des infrastructures modernes, notamment en matière de virtualisation, de cloud computing et de mobilité.

14 Conclusion

Le routage traditionnel est un moyen de gérer manuellement les chemins de communication dans un réseau informatique. Les principes fondamentaux du routage traditionnel, tels que les tables de routage et les protocoles tels que RIP et OSPF, ont été abordés dans ce chapitre. Même si le routage traditionnel offre un contrôle total et une prévisibilité, il présente des limites en termes de gestion et d'adaptabilité aux changements de réseau. Les approches plus dynamiques et automatisées, telles que le routage basé sur les protocoles de routage dynamique et l'utilisation de réseaux définis par logiciel (SDN), seront examinées dans les chapitres suivants.

Chapitre3:Comparaison entre le réseau traditionnel et le réseau SDN

1 Introduction

Les réseaux traditionnels et les réseaux SDN sont deux approches différentes pour la conception et la gestion des réseaux informatiques. Les réseaux traditionnels sont autonomes et gérés individuellement, alors que les réseaux SDN sont basés sur un modèle de contrôle centralisé géré par un logiciel sur un serveur centralisé. Ce chapitre présente une comparaison entre les deux approches ainsi qu'une présentation à l'émulateur Mininet et GNS3, qui permettent de créer un environnement de réseau virtuel pour tester les performances du réseau. Enfin, une analyse comparative des performances du réseau traditionnel et du réseau SDN est présentée à travers différents tests effectués avec l'émulateur Mininet et GNS3.

2 Comparaison entre le réseau traditionnel et SDN

	Réseau traditionnel	**SDN**
Configuration	Configuration manuelle sujette aux erreurs.	Configuration automatisée avec validation centralisée
Performance	Une quantité limitée d'informations est disponible et la configuration est relativement statique.	Échange d'informations inter-couches pour permettre un contrôle global dynamique.
Caractéristiques	La complexité de la gestion du réseau est augmentée par la nécessité de mettre en place un nouveau protocole pour chaque problème spécifique.	Une solution consiste à séparer les fonctions de traitement des données et de contrôle du réseau, et à permettre une programmabilité flexible pour chaque fonction.
Innovation	La mise en œuvre matérielle de nouvelles idées est difficile en raison de limitations dans l'environnement de test. De plus, le processus de normalisation est souvent long.	Les nouvelles idées peuvent être facilement mises en œuvre à l'aide de logiciels, car ces derniers sont plus flexibles que les solutions matérielles. De plus, les environnements de test sont suffisamment isolés pour éviter toute interférence avec

		les systèmes en production, permettant ainsi une validation complète des nouvelles idées. Enfin, le déploiement de ces idées peut être rapide grâce à l'amélioration des logiciels utilisés.

Tableau 3 : Tableau comparatife entre réseau traditionnel et réseau SDN

3 Outils de tests de performance

3.1 Définition de VirtualBox

Permettant ainsi d'exécuter plusieurs systèmes d'exploitation et leurs applications de manière isolée les uns des autres. En utilisant VirtualBox, vous pouvez installer et exécuter des systèmes d'exploitation tels que Windows, Linux, MacOs, Solaris, etc., sur un seul ordinateur, sans avoir besoin d'installer ces systèmes d'exploitation directement sur votre machine physique. Cela vous permet d'explorer différentes configurations, de tester des logiciels, d'isoler des environnements de développement, d'effectuer des tests de sécurité ET bien plus encore, sans affecter votre système d'exploitation principal. VirtualBox offre également des fonctionnalités avancées telles que le partage de fichiers entre l'hôte et l'invité, la gestion des réseaux virtuels et la possibilité de créer des instantanés pour capturer l'état d'une machine virtuelle à un moment donné.

3.2 Définition de GNS3

Le GNS3 est un simulateur graphique de réseaux qui nous permet de créer des topologies. De réseaux complexes et d'en établir des simulations. Ce logiciel est libre, et il est capable de faire fonctionner des images Cisco IOS comme si elles s'exécutaient sur de véritables équipements.

Pour fournir des simulations complètes et précises, GNS3 est fortement lié à :

- ***Dynamips*** : est un émulateur IOS Cisco.
- ***Qemu :*** est un émulateur de machine source et virtualiseur.
- ***VirtualBox*** : est un logiciel de virtualisation libre et puissant.

3.3 Présentation de Mininet

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. Mininet prend en charge la recherche, le développement, l'apprentissage, le prototypage, les tests, le débogage et toute autre tâche qui pourrait bénéficier d'un réseau expérimental complet sur un ordinateur portable ou un autre PC.

Via l'API Mininet

L'émulateur Mininet fournit une interface de programmation Python, l'une de ces utilisations consiste en la création de topologies customisées en précisant toutes les caractéristiques voulues des éléments du réseau, une approche pareille nous offre l'avantage.

Le fichier. Python doit être sauvegardé dans le répertoire /mininet/custom, de plus pour pouvoir.

Faire appel la topologie par l'option –custom.

Les commandes principales du code python pour créer la topologie réseau souhaitée sont les suivantes :

build () : La méthode à surcharger pour créer des topologies personnalisées.

addSwitch () : Ajoute un switch et retourne son nom.

addHost () : Ajoute un hôte et retourne son nom.

addLink () : Ajoute un lien bidirectionnel entre les composants passés en argument.

start () : Démarre le réseau.

Stop () : Arrête le réseau

4 Création de la topologie de réseaux traditionnels à l'aide de GNS3

Avant de créer la plateforme, il faut d'abord donner un nom au projet crée et le sauvegarder.

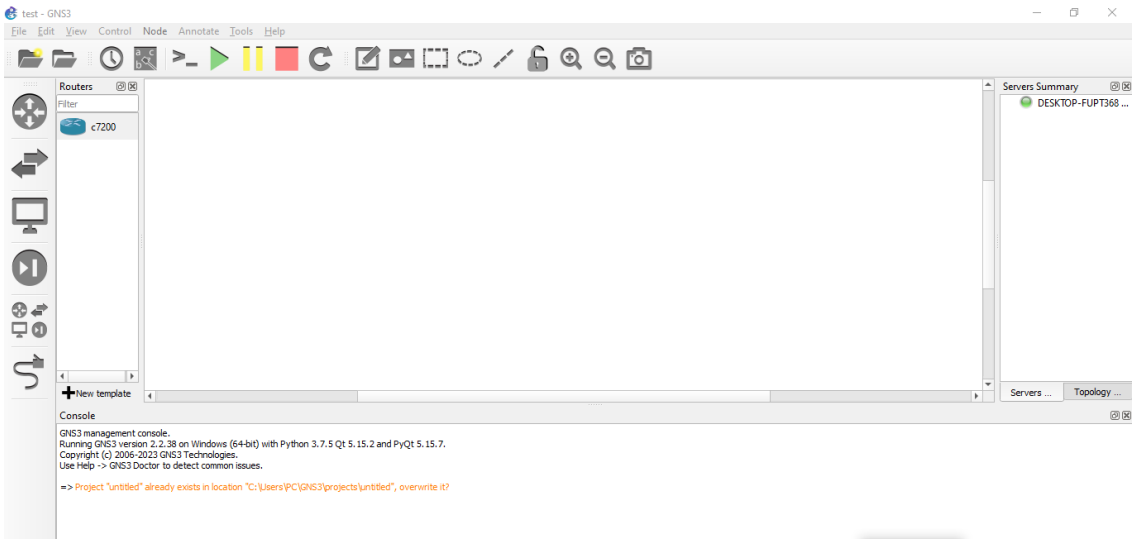


Figure 14:Interface du GNS3

4.1 Téléchargement des images IOS

Avant l'utilisation de n'importe quel équipement, il faut d'abord incorporer son image IOS correspondante, téléchargée du site officiel du GNS3.

En allant vers « Edit » ensuite « Preference » sur la barre d'outils.

Une fois sur « Preferences » on clique sur IOS Routers on clique sur le bouton « Brows » (parcourir), une interface comme celle au-dessous s'ouvre permettant de spécifier l'image depuis le répertoire dans lequel se trouve.

Dans cet exemple on prend le routeur C7200.

On ajoute les slots au routeur pour avoir des interfaces.

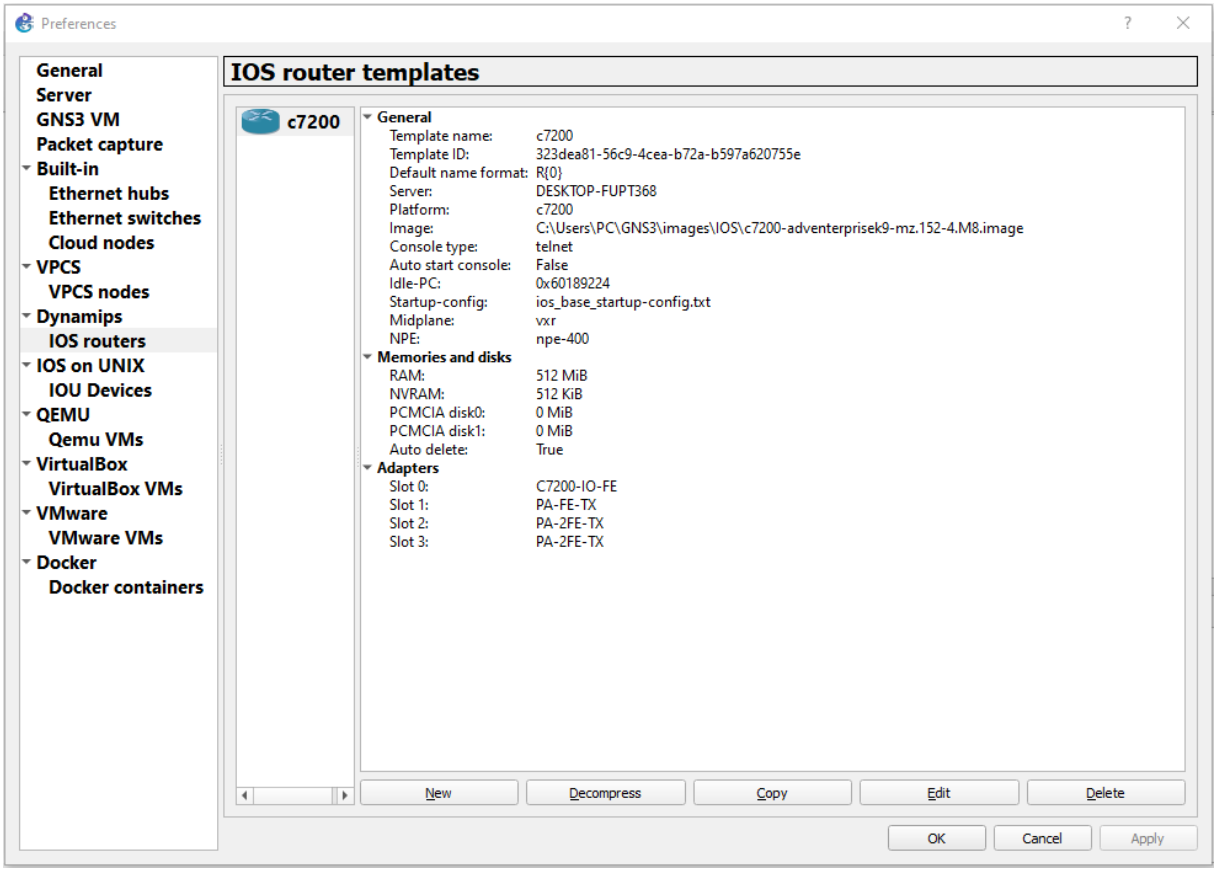


Figure 15: Installation image iso

4.2 Téléchargement des machines virtuelles

Pour accéder aux préférences, on se rendra dans la barre d'outils et on clique sur "Edit" suivi de "Preferences". Une fois dans les préférences, on sélectionne "VirtualBox VMs" et on clique sur le bouton "New" pour créer une nouvelle machine virtuelle. Dans cet exemple, nous choisirons un ordinateur virtuel basé sur Kali Linux. Ensuite, on ajoute les ordinateurs virtuels souhaités.

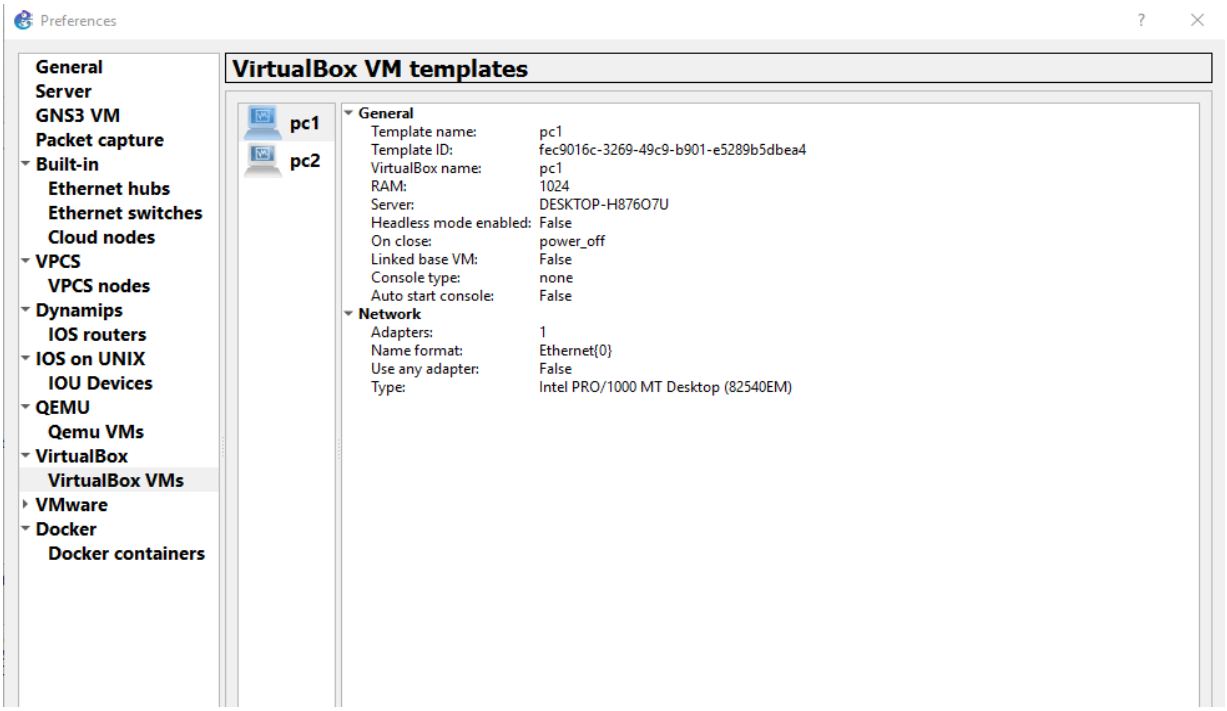


Figure 16: Installation de pc Virtual

4.3 Création de la topologie

Après avoir installé et configuré les images, on choisit les routeurs dont on aura besoin, il suffit de glisser le dessin situé à gauche et le déposer sur la fenêtre.

Après avoir formé le réseau, en plaçant les différents routeurs, sur la partie centrale on les relie par des câbles qui sont choisis à travers l'icône sur la barre de menu à gauche, de la figure Ce dernier devient utilisable et configurable.

Une fois que la plateforme est prête, on active la topologie avec l'icône sur la barre de menu en haut dans figure

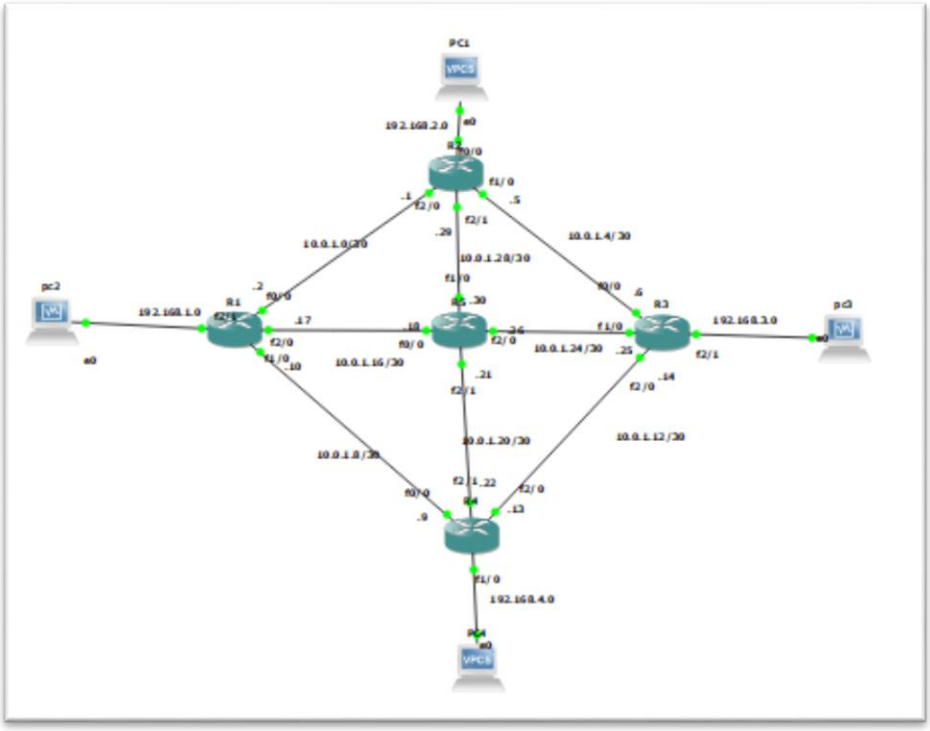


Figure 17: Topologie de réseau traditionnel dans l'Interface GNS3

4.4 Configuration des interfaces de router

On sélectionne chaque interface par la commande « interface fastEthernet X/Y ».

On attribue à chaque interface une adresse IP, en utilisant la commande « ip address » Suivie de l'adresse IP et masques de sous-réseaux, ensuite, à la fin on les active avec la commande « No shutdown ».

Vérification des interfaces utilisées après activation avec la commande « show ip interfaces brief » dans la console.

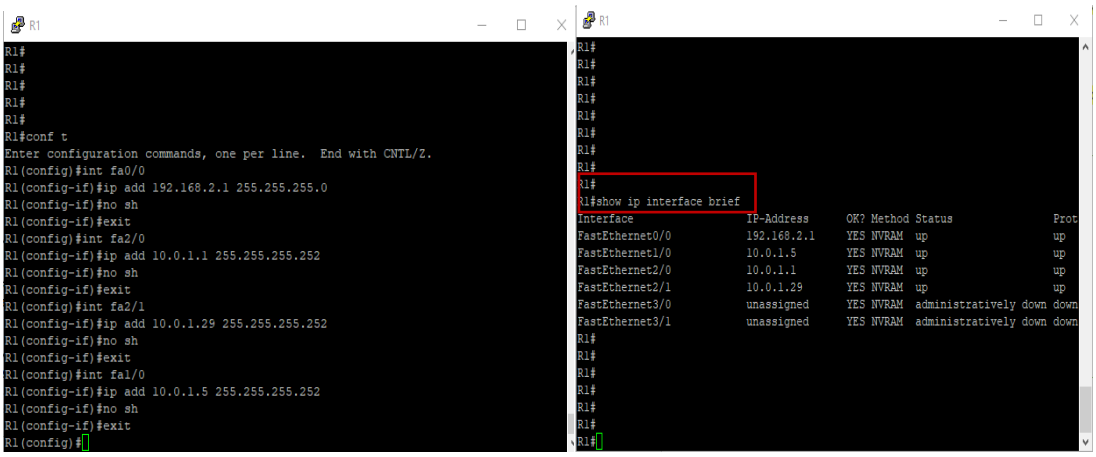


Figure 18: Configuration des interfaces

4.4.1 Configuration du protocole OSPF pour le router

L'activation du routage classique au niveau du backbone, c'est-à-dire entre les PE-Routeurs et les P-Routeurs., Nous avons portés notre choix sur le protocole OSPF à cause de ses multiples avantages :

- C'est un protocole de routage à états de liens.
- Il est rapide en termes de convergence.
- La configuration d'OSPF doit être effectuée sur tous les routeurs du réseau Comme suit :

On active pour chaque routeur le protocole OSPF, qui permet de créer une table de routage Dans chaque routeur, avec les commandes suivantes :

- « router ospf » : pour l'activation du processus ospf, le fig. 19 ci-dessus représente l'identifiant du routeur.
- « network » : pour déclarer et spécifier le réseau participant au processus ospf.
- « exit » : pour sortir du mode configuration.

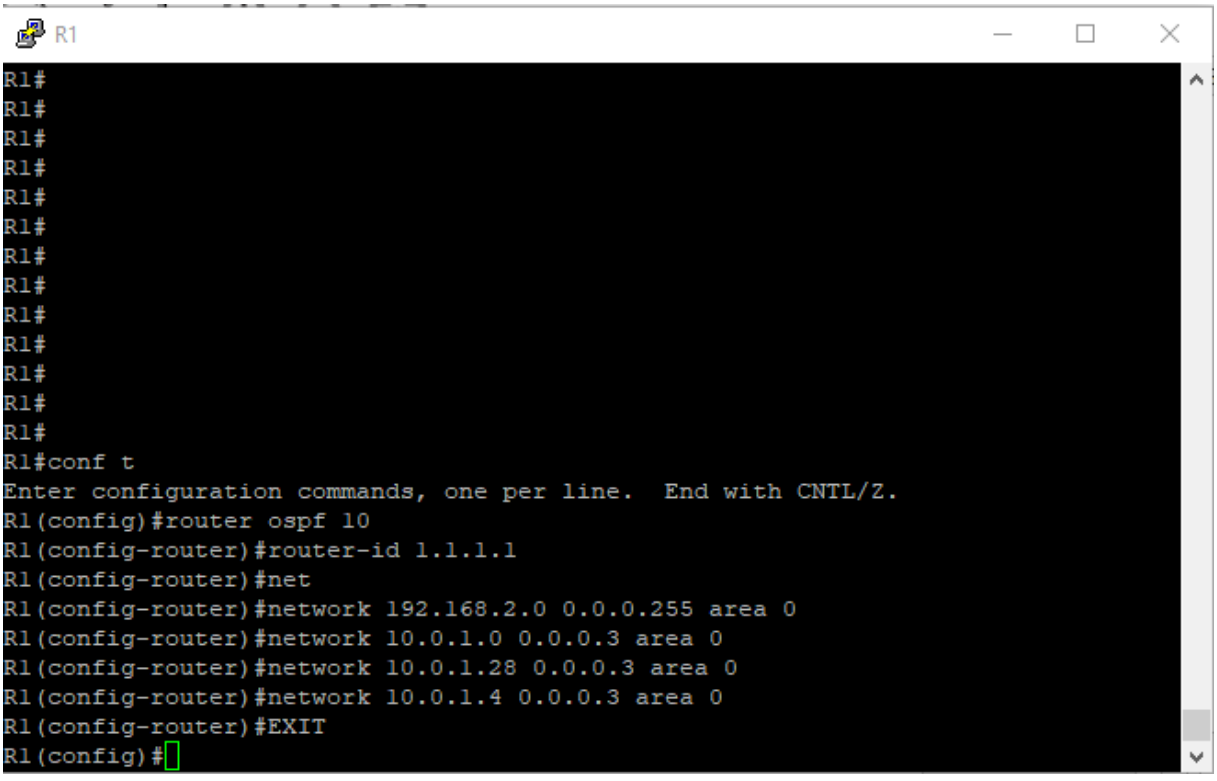


Figure 19:Activation d'ospf

Après la configuration et pour tester le bon fonctionnement du protocole OSPF, on Exécute la commande « show ip route OSPF » qui nous montre la table de routage

-La lettre « O » représente les liens connectés par le protocole OSPF.

-La table de routage OSPF est ci-dessous

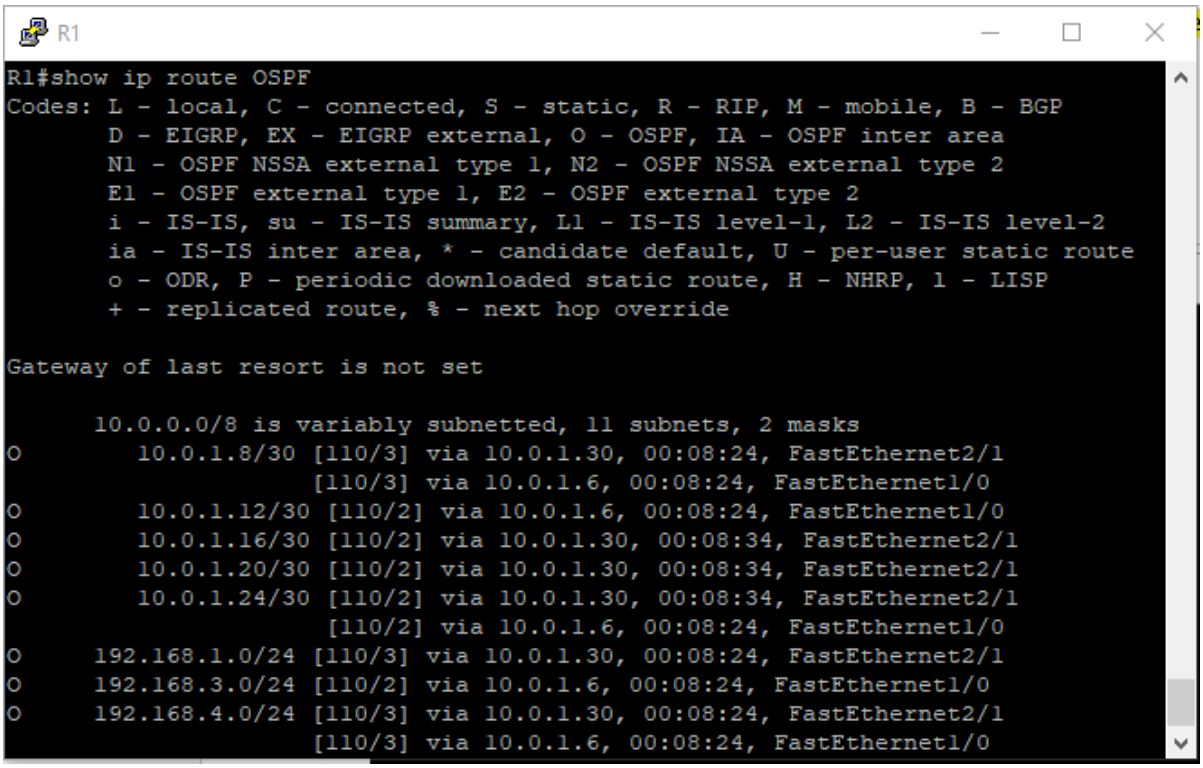


Figure 20: Les routes d'ospf

4.5 Configuration des pc

Etape 1 :

Installation de Kali linux :

-Téléchargement de Kali Linux : Sur le site officiel de Kali Linux (https://www.kali.org/downloads/) et l'image ISO correspondante à votre système 64 bits. Choisir la version appropriée pour votre système hôte (Windows 10).

-Création d'une machine virtuelle : Lancez VirtualBox et cliquez sur le bouton "Nouvelle" pour créer une nouvelle machine virtuelle. Donnez un nom à votre machine virtuelle Kali Linux, sélectionnez le type de système d'exploitation Linux et la version Debian. Cliquez sur "Suivant".

-Attribution de la mémoire RAM : Sélectionnez la quantité de mémoire RAM que vous souhaitez allouer à la machine virtuelle. Il est recommandé d'allouer au moins 1 Go de RAM pour une utilisation optimale de Kali Linux. Cliquez sur "Suivant".

-Création d'un disque dur virtuel : Sélectionner "Créer un disque dur virtuel maintenant" et cliquer sur "Suivant". Choisissez le type de fichier de disque dur virtuel et cliquez sur "Suivant". Sélectionnez l'allocation de stockage dynamique pour économiser de l'espace sur votre disque dur physique, puis choisissez la taille du disque dur virtuel. Cliquez sur "Créer".

-Configuration de la machine virtuelle : Dans la fenêtre principale de VirtualBox, cliquez avec le bouton droit de la souris sur la machine virtuelle que vous venez de créer Kali Linux et

sélectionnez "Configuration". Dans les paramètres de la machine virtuelle, allez dans la section "Stockage" et cliquez sur l'icône en forme de CD/DVD. Sélectionnez l'option "Choisir un fichier de disque optique virtuel" et sélectionnez l'image ISO de Kali Linux que vous avez téléchargée. Cliquez sur "OK" pour fermer la fenêtre de configuration.

-Installation de Kali Linux : Démarrez la machine virtuelle en cliquant sur le bouton "Démarrer". Kali Linux se lancera à partir de l'image ISO que vous avez montée. Choisir entre l'installation en mode graphique, selon vos préférences. Suivez les instructions à l'écran pour installer Kali Linux sur le disque dur virtuel.

-Configuration de Kali Linux : Une fois l'installation terminée, Kali Linux redémarrera. Vous pouvez maintenant vous connecter à votre machine virtuelle Kali Linux en utilisant le nom d'utilisateur (user) et le mot de passe (user) que vous avez définis lors de l'installation.

-Etape 2 :

Dans le GNS3 démarrer la machine et l'ordinateur virtuel et accédons au terminal en tant que super utilisateur (root). Nous attribuons à chaque ordinateur virtuel une adresse IP en utilisant la commande "ifconfig (adresse IP) + netmask (masque de sous-réseau) + broadcast". Nous attribuons la passerelle en utilisant la commande "route add default gw (adresse IP de la passerelle) eth0". Nous vérifions les adresses utilisées après leur activation en utilisant la commande "ifconfig" dans le terminal.

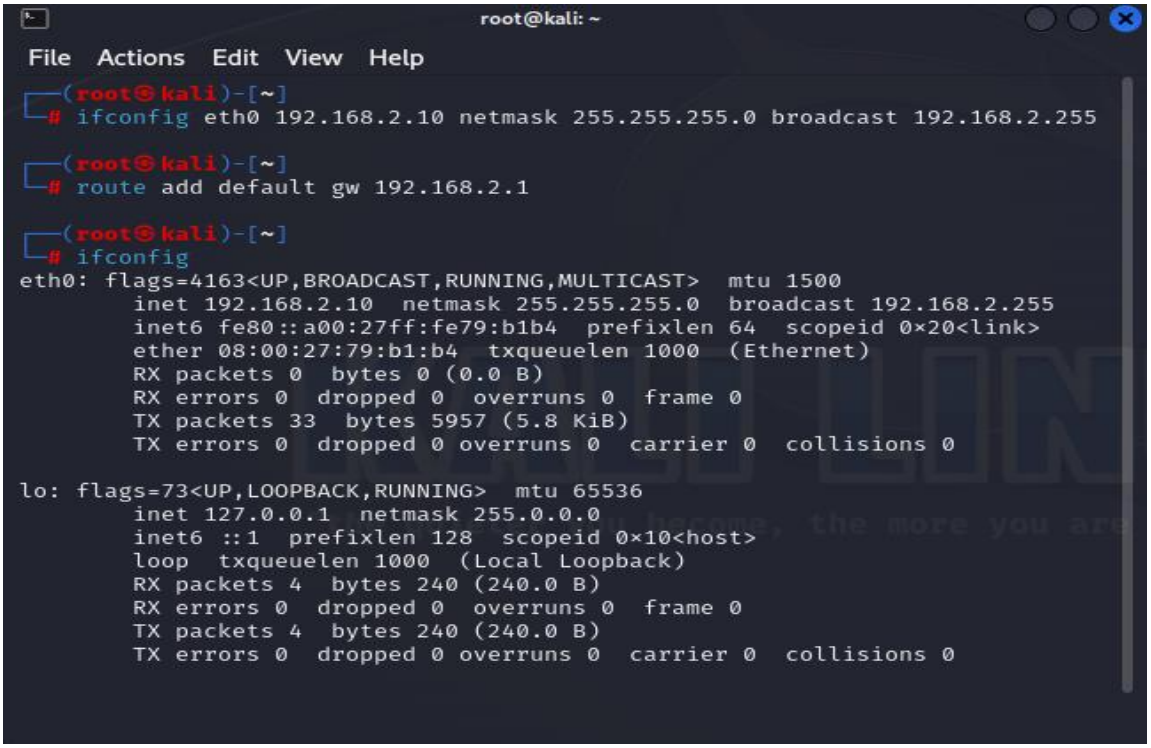


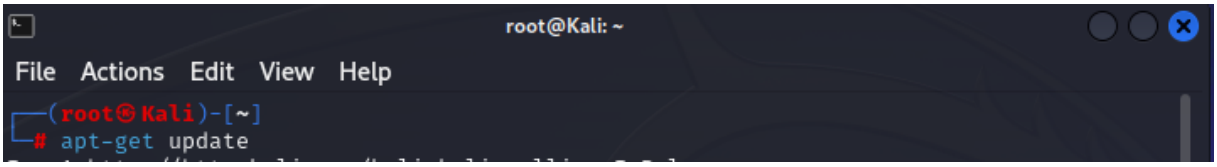
Figure 21:Configuration de adresse ip de pc

-Etape 3:

Installation d'iperf3 dans la machine virtuelle :

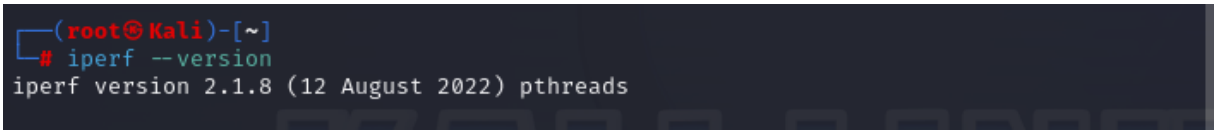
-***Ouvrez un terminal*** : cliquez ver le terminal ou Cliquez sur l'icône du terminal dans la barre des tâches ou utilisez le raccourci clavier Ctrl+Alt+T pour ouvrir un terminal.

-***Mettez à jour les dépôts*** : Avant d'installer iperf3, il est recommandé de mettre à jour les informations sur les paquets disponibles dans les dépôts. Exécutez la commande suivante dans le terminal:



-***Installer iperf3*** : Utilisez la commande suivante pour installer iperf3 :

-***Vérifier l'installation*** : Après l'installation, on peut vérifier si iperf3 est bien installé en exécutant la commande suivante :



-Cela affichera la version d'iperf3 installée sur votre système.

4.6 Evaluation des performances des réseaux traditionnels

4.6.1 Le temps de réponse

Ping mesure le temps d'aller-retour pour les messages envoyés de l'hôte d'origine à un hôte de destination qui sont renvoyés à la source, il fonctionne en envoyant des paquets de demande d'écho ICMP à l'hôte cible et en attendant une réponse d'écho ICMP.

La ligne de commande utilisée pour mesurer le temps de réponse est la suivante :

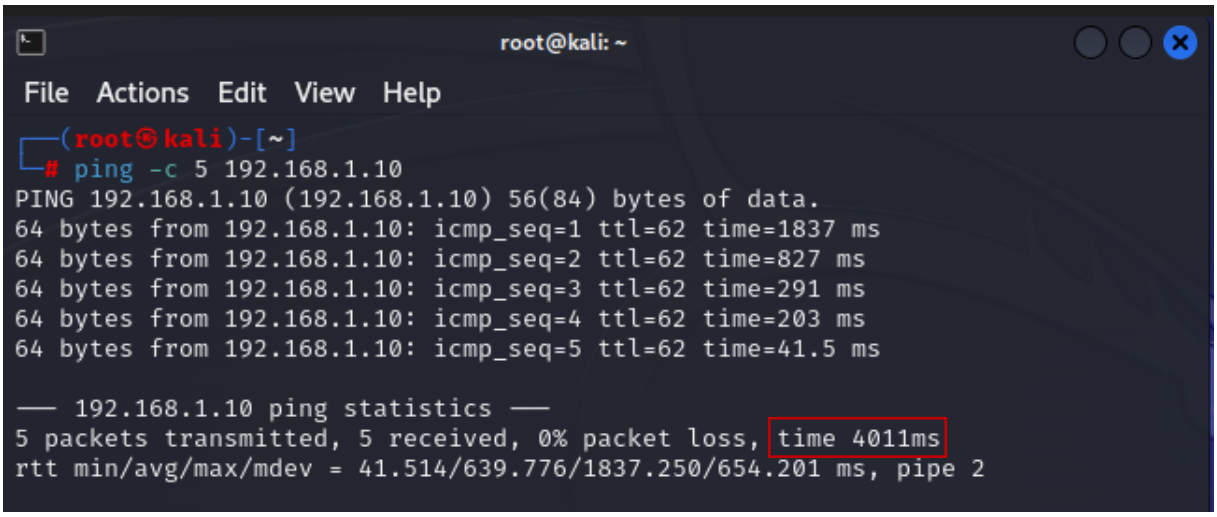
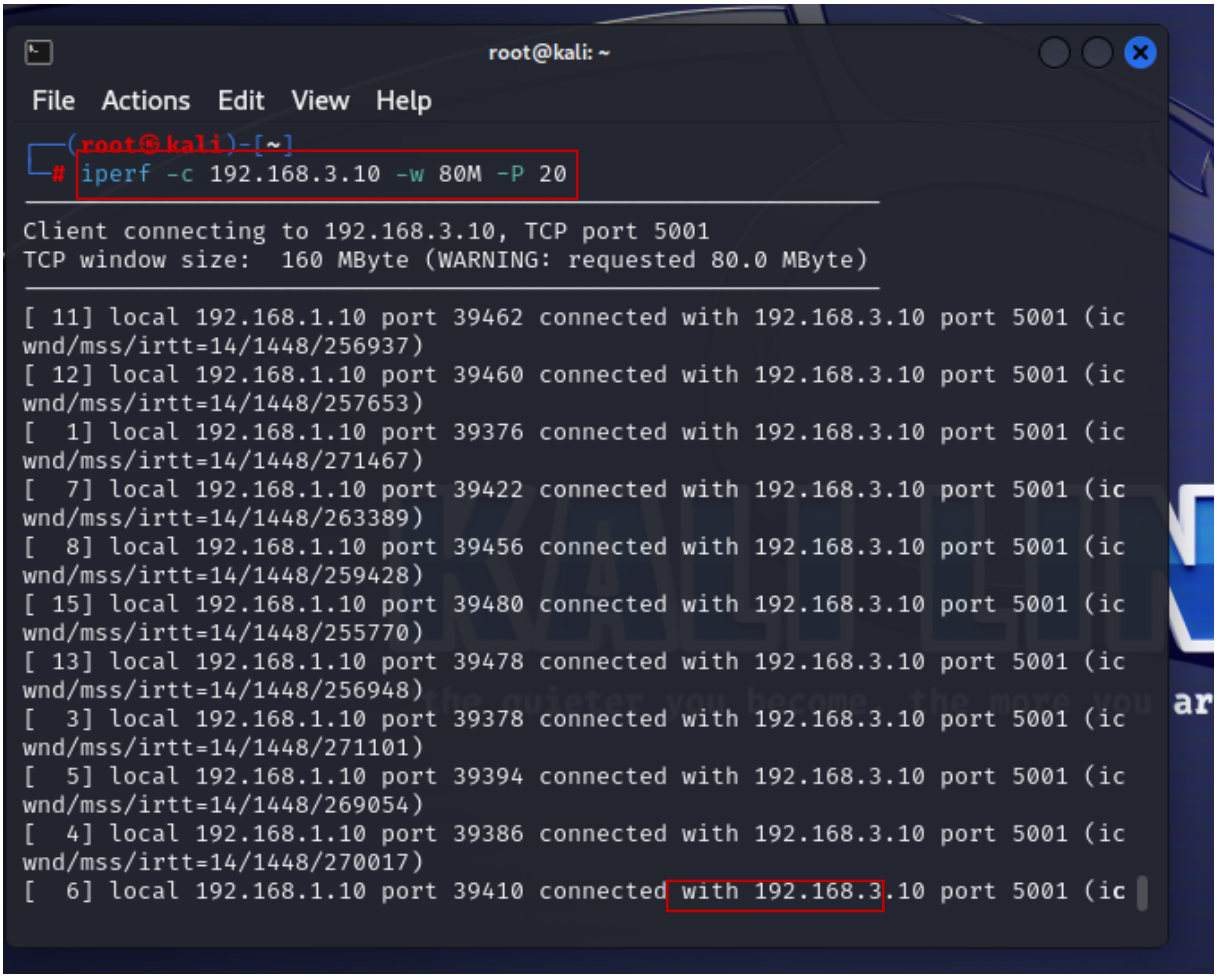


Figure 22:Ping réussi entre « h1 » et « h2 ».

Le temps de réponse de 5 paquets en ICMP entre les deux hosts est de 4011 ms.

4.6.2 La bande passante

Nous avons testé la performance du débit avec la commande suivante :



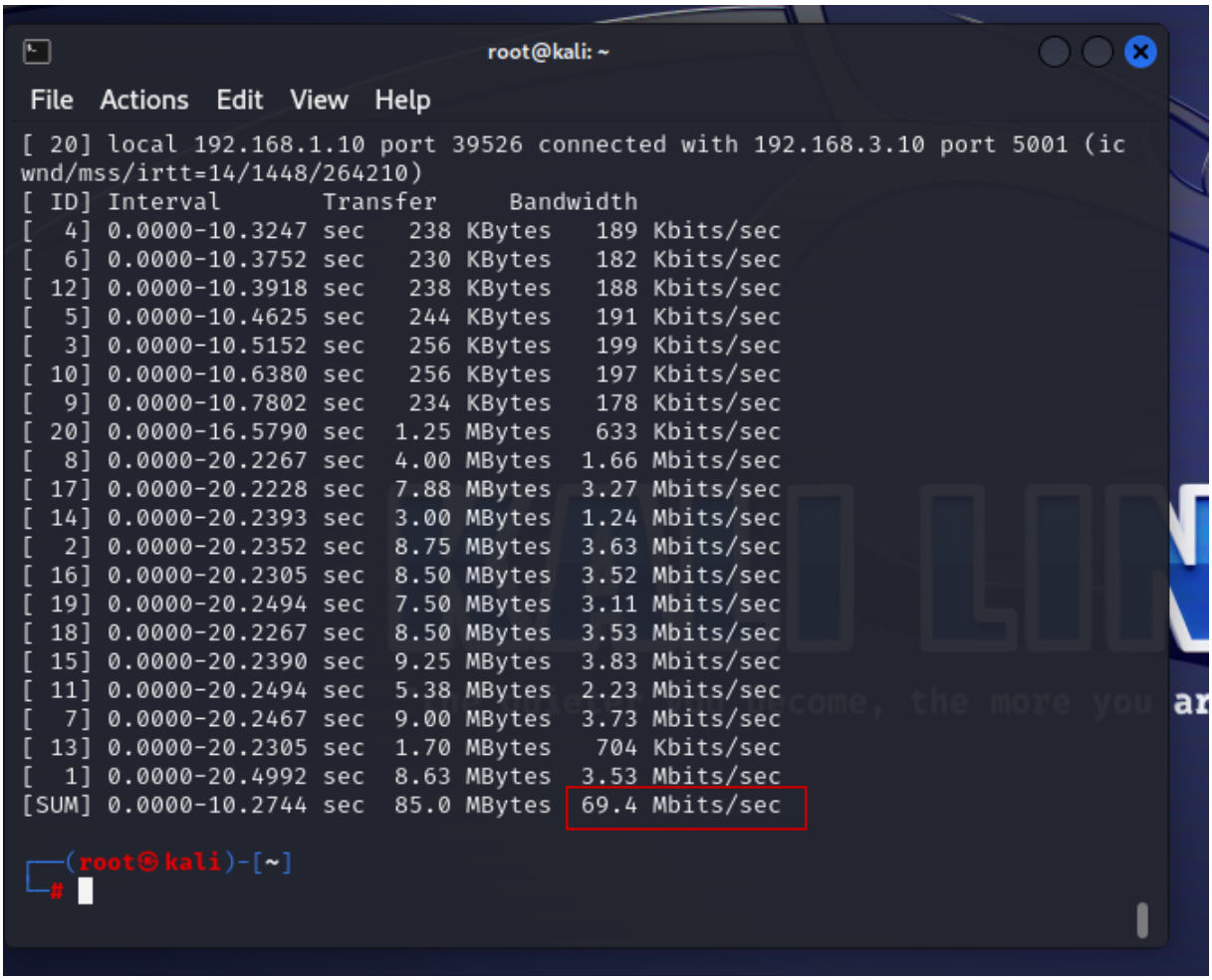


Figure 23:Test de la largeur de bande.

Le débit mesurer entre les deux hôtes est de 69.4 Mbits/s

5 Création de la topologie SDN à l'aide de Mininet:

5.1 Installation de Controller OpenDaylight

Les étapes d'installation et de configuration d'OpenDaylight sont :

Étape 1 : pour installer machine virtuelle, je me suis levé et installé VMware.

-Créez une machine virtuelle avec Ubuntu comme système d'exploitation (Ubuntu 16.04.2).

-Allouez une quantité appropriée de mémoire vive (RAM) à la machine virtuelle.

-Créez un disque dur virtuel de type "VDI" avec une allocation dynamique.

-Et mettre le réseau de la machine Virtual Bridge pour compter à internet.

Étape 2 : Installation de Java et d'OpenDaylight

Ouvrir la machine virtuelle et connectez-vous avec le nom d'utilisateur "odl" et le mot de passe "odl".

Mettre à jour les paquets en exécutant la commande "apt-get update" dans le terminal.

```
root@odl :~# apt-get update
```

-Installer l'environnement d'exécution Java, Maven, Git et les versions 8 de JDK et JRE avec la commande suivante :

```
root@odl :~# apt-get install maven git openjdk-8-jre openjdk-8-jdk unzip
```

-Télécharger et installer OpenDaylight avec les commandes suivantes :

```
root@odl :~# wget
https://nexus.opendaylight.org/content/repositories/public/org/opendaylight/integratio
n/distribution-karaf/0.6.2-Carbon/distribution-karaf-0.6.2-Carbon.zip
root@odl :~# unzip distribution-karaf-0.6.2-Carbon.zip
```

-Accéder au répertoire d'installation d'OpenDaylight avec la commande suivante :

```
root@odl :~# cd distribution-karaf-0.6.2-Carbon/bin
```

-Définisser la variable d'environnement JAVA_HOME en utilisant la commande suivante :

```
root@odl:~ / distribution-karaf-0.6.2-Carbon/bin# exportJAVA_HOME=/usr/lib/jvm/java-
1.8.0-openjdk-amd64
```

Démarrer le contrôleur OpenDaylight en exécutant la commande suivante :

```
root@odl :~ /distribution-karaf-0.6.2-Carbon/bin#./karaf
```

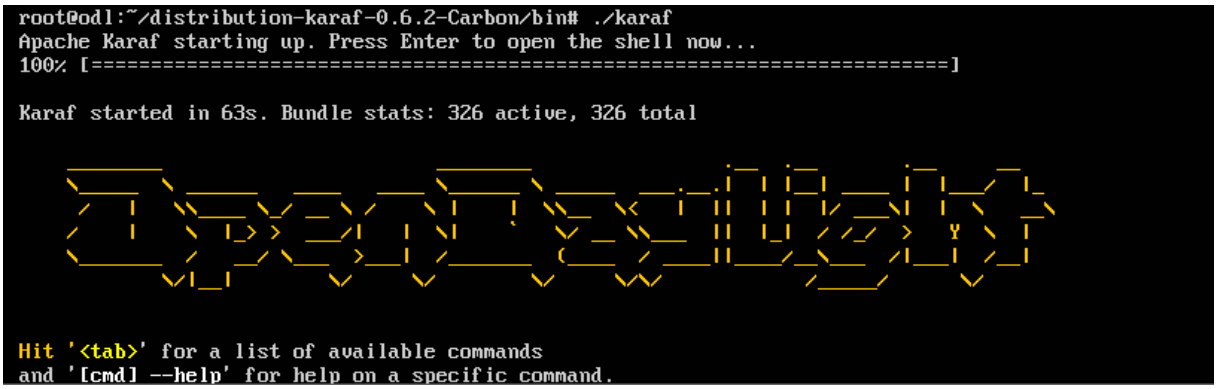


Figure 24:Démarrage de contrôleur OpenDaylight.

Étape 3 : Installation des fonctionnalités d'OpenDaylight

-Dans le terminal où le contrôleur OpenDaylight est en cours d'exécution, installez les fonctionnalités nécessaires en utilisant les commandes suivantes :

```
Opendaylight-user@root > feature:install odl-restconf
Opendaylight-user@root > feature:install odl-l2switch-switch
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
```

Étape 4: Obtenir l'adresse IP de la machine OpenDaylight

-Dans le terminal, exécuter la commande "ifconfig" pour obtenir l'adresse IP de la machine virtuelle OpenDaylight.

```
odl@odl:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:c1:3e:2d
          inet addr:192.168.211.77  Bcast:192.168.211.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec1:3e2d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:656 errors:0 dropped:0 overruns:0 frame:0
          TX packets:656 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:50048 (50.0 KB)  TX bytes:50048 (50.0 KB)

odl@odl:~$
```

Figure 25:Adresse ip de contrôler.

Étape 5: Accéder à l'interface graphique d'OpenDaylight.

-Dans n'importe quel navigateur Web dans notre cas utilisé Microsoft Edge, utilisez l'URL suivante: "http ://192.168.211.77 :8181/index.html".

Utiliser les identifiants par défaut pour vous connecter à l'interface :

Nom d'utilisateur : admin.

Mot de passe : admin.

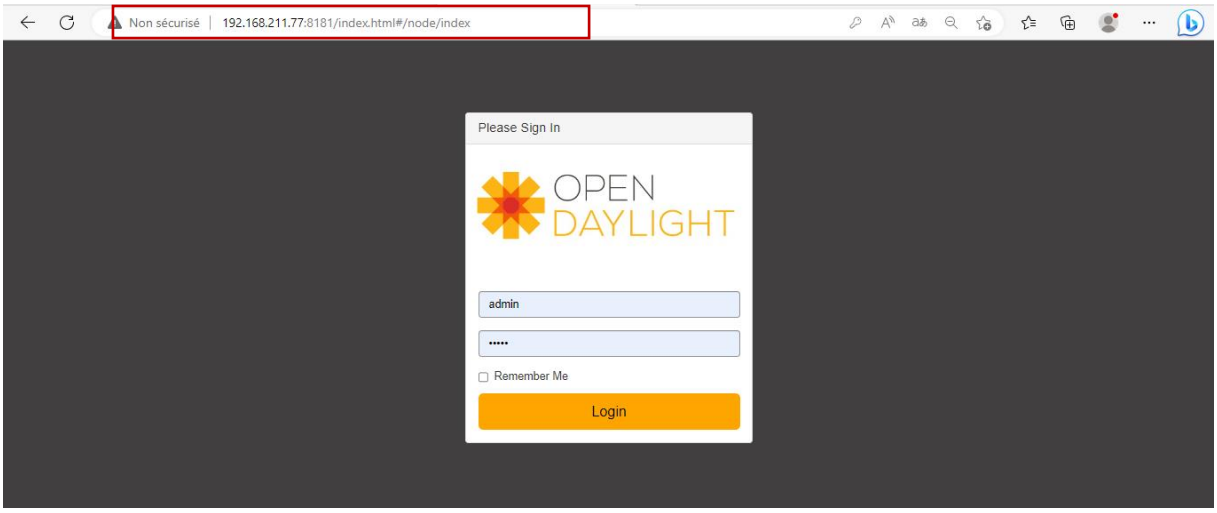


Figure 26:Interface login pour Controller.

5.2 Création de la topologie avec Mininet

Apres le téléchargement de l'image Mininet et le programme de virtualisation VirtualBox :

-Connection à VirtualBox et joindre le fichier de l'image Mininet

-Lancement du VMMininet une fois qu'elle est activée

- connection à Mininet à l'aide d'un nom d'utilisateur et un mot de passe :

Login: Mininet

Password: Mininet

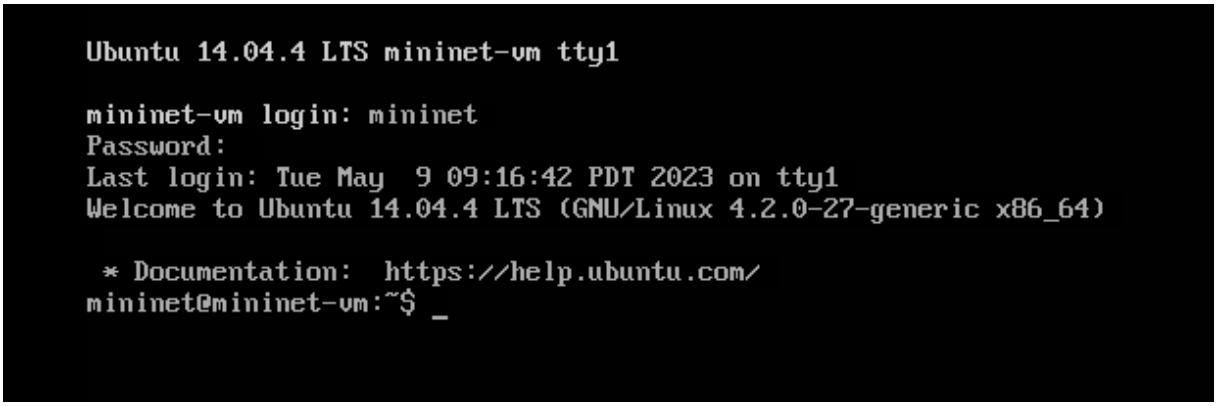


Figure 27: Login de Mininet

Celui-ci qui va créer la topologie propre, avec la commande suivante :

```
$Sudo nano stp.py
```

Figure 28: Création d'une topologie avec python dans Mininet.

Afin de créer la topologie on doit ouvrir un terminal de la machine virtuelle, puis on démarre l'émulateur mininet, celui-ci qui va créer la topologie propre, avec la commande suivante :

```
$Sudo mn -- custom=stp .py --topo=mytopo --controller=remote,ip=192.168.211.77
```

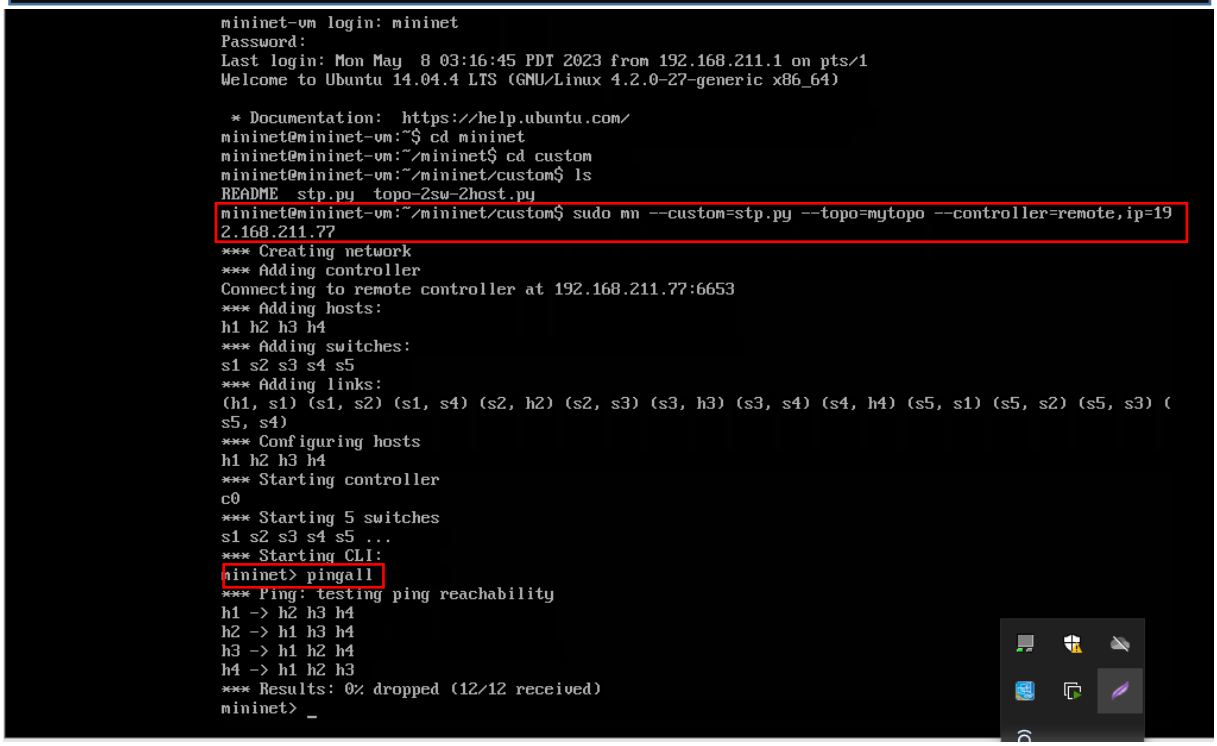


Figure 29:Création d'une topologie avec Mininet.

Une fois la topologie créée avec mininet, tout le détail sera affiché dans la page d'accueil de OpenDaylight comme montre la figure

La topologie est affichée par le contrôleur OpenDaylight, Cette topologie contient :

- Commutateur.
- Hôtes.
- Liens qui relient les commutateurs et les hôtes entre eux.

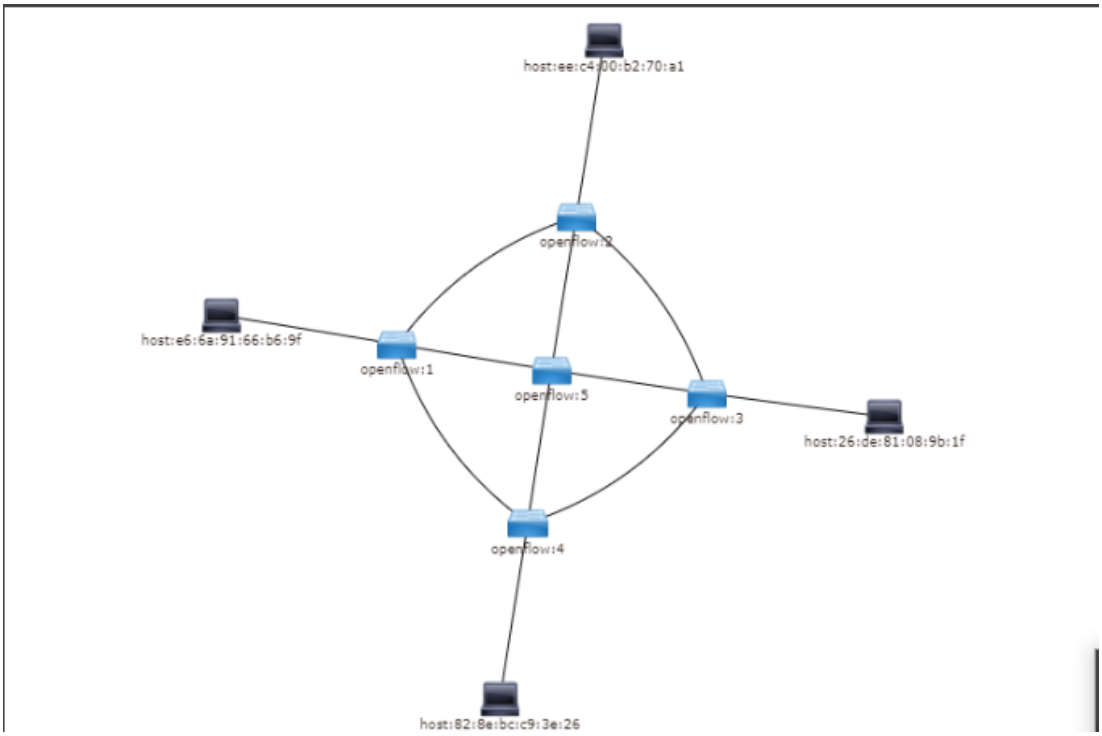


Figure 30: Topologie de SDN l'Interface web de l'OpenDaylight.

5.3 Evaluation des performances de la topologie

5.3.1 Le temps de réponse

Mesure le temps d'aller-retour pour les messages envoyés de l'hôte d'origine à un hôte de destination qui sont renvoyés à la source, il fonctionne en envoyant des paquets de demande d'écho ICMP à l'hôte cible et en attendant une réponse d'écho ICMP.

La ligne de commande utilisée pour mesurer le temps de réponse est la suivante :

```
mininet> h1 ping -c 5 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.498 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.410 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.319 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.249 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.427 ms

--- 10.0.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 0.249/0.380/0.498/0.089 ms
mininet>
```

Figure 31: Mesure du temps de réponse.

Le temps de réponse de 5 paquets en ICMP entre les deux hosts est de 4003 ms.

5.3.2 La bande passante

Nous avons testé la performance du débit avec la commande suivante :

```
mininet> iperf h1 h3
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['170 Mbits/sec', '174 Mbits/sec']
mininet>
```

Figure 32: Test de performance du débit.

Le débit mesurer entre les deux hôtes est de 170 Mbits/s

6 Tableau des résultats

	Temps de réponse	La bande passante
Réseau traditionnel	4011 ms.	69.4 Mbits/s
Réseau SDN	4003 ms.	170Mbit/s

Tableau 4 :Comparaison entre le temps de réponse et le débit dans réseaux traditionnels et SDN

7 Analyse des résultats et discussion

Nous allons mener une étude comparative pour évaluer les performances d'un réseau traditionnel par rapport à un réseau SDN en ce qui concerne le temps de réponse et la bande passante. Ces deux aspects sont essentiels pour évaluer l'efficacité et la rapidité des communications au sein d'un réseau. Nous allons examiner les valeurs spécifiques de chaque réseau afin de déterminer leurs avantages et leurs différences.

En ce qui concerne la bande passante, dans notre exemple, le réseau SDN affiche un débit mesuré de 170 Mbits/s et un débit maximal de 200 Mbits/s, tandis que le réseau traditionnel atteint un débit mesuré de 69,4 Mbits/s et un débit maximal de 100 Mbits/s.

Pour calculer le pourcentage de bande passante utilisée, nous utilisons la formule suivante :

Pourcentage de bande passante = (Débit mesuré / Débit maximal) * 100

Pour le réseau SDN :

Pourcentage de bande passante = (170 / 200) * 100 = 85

Donc, dans ce cas, le réseau SDN utilise 85 % de sa bande passante maximale.

Pour le réseau traditionnel :

Pourcentage de bande passante = (69,4 / 100) * 100 = 69,4

Dans ce cas, le réseau traditionnel utilise 69,4 % de sa bande passante maximale.

Ces résultats indiquent que le réseau SDN utilise une plus grande proportion de sa bande passante maximale par rapport au réseau traditionnel dans cet exemple. Cela peut être attribué à l'optimisation de la gestion du trafic dans les réseaux SDN, ce qui permet une meilleure utilisation des ressources disponibles et une répartition plus efficace du trafic. Par conséquent, les réseaux SDN peuvent offrir des débits bien plus élevés que les réseaux traditionnels dans notre exemple.

En ce qui concerne le temps de réponse, dans notre exemple, le réseau SDN affiche un temps de réponse de 4003 ms, tandis que celui du réseau traditionnel est de 4011 ms. Les temps de réponse sont relativement similaires dans ce cas spécifique. Il est important de noter que les valeurs de temps de réponse peuvent varier en fonction de plusieurs facteurs, tels que la complexité des opérations de routage et de commutation, la taille du réseau, la charge de trafic, etc. En général, les réseaux SDN ont le potentiel d'offrir des temps de traitement plus courts grâce à la centralisation de la gestion du trafic et à une prise de décision plus efficace concernant le routage des paquets

En analysant les performances de manière comparative, nous constatons des différences significatives entre un réseau traditionnel et un réseau SDN en termes de bande passante. Bien que le temps de réponse mesuré dans le réseau SDN soit légèrement inférieur à celui du réseau traditionnel, il n'y a pas de différence majeure en termes de délais de transmission. Cependant, la bande passante mesurée dans le réseau SDN est nettement supérieur, offrant une capacité de transmission des données considérablement améliorée par rapport au réseau traditionnel. Il est important de noter que ces mesures spécifiques fournissent une comparaison limitée entre les deux types de réseaux. D'autres aspects tels que la flexibilité, la gestion centralisée, la programmabilité et la sécurité doivent également être pris en compte pour une évaluation complète des performances et des avantages d'un réseau SDN par rapport à un réseau traditionnel. Cependant, les résultats indiquent que le réseau SDN offre des améliorations potentielles en termes de débit, ce qui peut entraîner une meilleure expérience utilisateur et des communications plus efficaces dans un réseau.

8 Conclusion

Dans ce chapitre, nous avons utilisé des outils tels que Mininet et GNS3 pour mener notre étude comparative entre le réseau traditionnel et le réseau SDN. Nous avons réussi à atteindre notre objectif principal qui était d'analyser et de comparer ces deux types de réseaux. Grâce à gns3, nous avons pu créer des topologies de réseaux virtuels et simuler le fonctionnement d'un réseau traditionnel. Nous avons examiné les protocoles de routage traditionnels tels qu'OSPF. En utilisant mininet, nous avons pu explorer le réseau SDN et ses caractéristiques distinctives.
Cette étude comparative nous a permis de mieux comprendre les différences entre le réseau traditionnel et le réseau SDN. Elle souligne l'importance de considérer les besoins spécifiques de l'infrastructure et de l'organisation lors du choix entre ces deux types de réseaux.

Conclusion générale

Enfin, ce mémoire de fin d'étude nous a permis d'examiner et comparer les réseaux traditionnels et les réseaux SDN. Nous avons pu évaluer les avantages et les inconvénients de chaque méthode ainsi que leurs effets sur la gestion et les performances des réseaux grâce à une analyse approfondie.

Nos résultats ont montré que les réseaux traditionnels offrent une infrastructure robuste et bien établie, mais souffrent souvent de limitations en termes de scalabilité, de flexibilité et de gestion complexe. Les réseaux SDN, en revanche, offrent une approche innovante basée sur la division du plan de contrôle et du plan de données, ce qui permet une programmabilité plus fine, une automatisation accrue et une gestion du réseau centralisée. Cela offre des avantages considérables en termes d'agilité, de réactivité et de facilité de mise en œuvre de nouveaux services.

Notre étude met en évidence le potentiel des réseaux SDN pour changer la gestion des réseaux et répondre aux besoins croissants de flexibilité et d'agilité de l'industrie.

Nous recommandons aux utilisateurs et aux professionnels des réseaux de mener une évaluation approfondie des avantages et des inconvénients des réseaux traditionnels et des réseaux SDN en fonction de leurs besoins spécifiques. Pour une transition plus fluide et une exploitation optimale des avantages des réseaux SDN, il est possible de considérer une approche hybride ou progressive qui combine les éléments des deux approches.

Bibliographie

[1]D. Kreutz, F.M.V. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky et S. Uhlig, "Software-defined networking: A comprehensive survey", Proceedings of the IEEE,IEEE xplore, 103(1), pp. 14–76, 2015.

[2]N. Feamster, J. Rexford et E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks", ACM SIGCOMM Computer Communication Review, 44(2), pp. 87–98, 2014

[3]A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan et H. Zhang, "A Clean Slate 4D Approach to Network Control and Management", ACM SIGCOMM Computer Communication Review, 35(5), pp. 41–54, 2005.

[4]M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown et S. Shenker, "Ethane: Taking Control of the Enterprise", Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 1–12, 2007.

[5]S. Ben Chahed, « Mise en oeuvre des aspects de gestion des réseaux définis par logiciels (réseaux SDN) », masters, École Polytechnique de Montréal, 2015

[6]M. Casado, F. Foster, and A. Guha, "Abstractions for softwaredefined networks. Commun". ACM, Sep. 2014

[7]P. Oppenheimer, "Top Down Network Design", Cisco Press, 2011.

[8] T. Tsou, P. Aranda, H. Xie, R. Sidi, H. Yin, and D. Lopez, "SDNi: A MessageExchange Protocol for Software Defined Networks (SDNS) across Multiple Domains", [En ligne] Disponible:https://tools.ietf.org/html/draft-yin-sdn-sdni-00, 2012

[9]« ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN) - ACM SIGCOMM 2014 ». [En ligne]. Disponible sur: https://conferences.sigcomm.org/sigcomm/2014/hotsdn.php. [Consulté le: 28-août-2019].

[10]K. Phemius, M. Bouet, et J. Leguay, « DISCO: Distributed multi-domain SDN controllers », in 2014 IEEE Network Operations and Management Symposium (NOMS), 2014, p. 14.

[11]B. Lee, S. H. Park, J. Shin, et S. Yang, « IRIS: The Openflow-based Recursive SDN controller », in 16th International Conference on Advanced Communication Technology, 2014, p. 1227 1231.

[12]Z. Cai, A. L. Cox, et T. S. E. Ng, « Maestro: A System for Scalable OpenFlow Control », déc. 2010.

[13]J. Medved, R. Varga, A. Tkacik, et K. Gray, « OpenDaylight: Towards a Model-Driven SDN Controller architecture », in Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, 2014, p. 16.

[14]N. Gude et al., « NOX: Towards an Operating System for Networks », SIGCOMM Comput Commun Rev, vol. 38, no 3, p. 105–110, juill. 2008.

[15]S. Kaur, J. Singh, ET N. Singh Ghumman, « Network Programmability Using POXController », 2014.

[16]H. Akçay ET D. Yiltas, « Web-Based User Interface for the Floodlight SDN Controller », Int. J.Adv. Netw. Appl., vol. 8, p. 3175 3180, avr. 2017.

[17]C. El Khalfi, A. El Qadi, ET H. Bennis, « A Comparative Study of Software Defined Networks Controllers », 2017, p. 15.

[18]15. Goransson, P., Black, C., & Culver, T. (2016). Software Defined Networks A Comprehensive Approach. 2éme édition. New York, USA. : Elsevier Science

[19]https://www.opennetworking.org/images/stories/downloads/sdnrsources/onf-specifications/openflow/openflow-spec-v1. 4.0. pdf, accessed on 23/3/2014.

[20]N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, March 2008.

[21]IBM, Software-Defined Networking: A New Paradigm for Virtual, Dynamic, Flexible Networking, October 2012

[22] T. D. Nadeau and K. Gray, "SDN: Software Defined Networks". O'Reilly, August 2013.

[23] B. Pfaff, B. Lantz, B. Heller et al., OpenFlow Switch Specification, 2012.

[24]OpenFlow Switch Consortium, OpenFlow Spec, v1.3.0

[25]V. W. Protocol, "Open Flow Switch Specification," vol. 0, 2009.

[26]V. Implemented and W. Protocol, "Open Flow Switch Specification," 2011.

[27]V. W. Protocol, "Open Flow Switch Specification," vol. 0, pp. 0–105, 2012.

[28]V. W. Protocol, "Open Flow Switch Specification," vol. 0, pp. 1–206, 2013.

[29]V. Protocol, "Open Flow Switch Specification" vol. 0, 2014.

[30]Mémoire de Fin d'Etudes De MASTER ACADEMIQUE Filière :
Télécommunications Spécialité : Réseaux et Télécommunications Présenté par : Mr AICHAOUI Anis Mr AITBELKACEM Yanis.
[31]Mémoire de fin d'étude de master académique présenté par MERIDJI Rania.
[32]Routageindirecthttp://www.iro.umontreal.ca/~kropf/ift6052/exercices/applets/applet5/introduc.h tmconsulté le 06 aout 2020.
[33]Modèle OSI et routage : http : //benabdellah-informatique.wifeo.com/Modle-OSIetroutage.pdf consulté le 03 septembre 2020.
[34] Mémoire fin d'étude de master, Les réseaux tolérants aux délais : le routage et les problèmes d'acheminement.
[35] Radia Perlman, « Interconnexions Ponts et routeurs », page 389-397.