REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA BOUMERDES

FACULTE DE TECHNOLOGIE

Département : Ingénierie des systèmes électriques

Filière : Télécommunications

Mémoire de projet de fin d'études pour l'obtention du

Diplôme de Master

Spécialité : Réseaux et Télécommunications

Thème

Système automatisé contre le Détournement des préfixes BGP « ARTEMIS »

Etudié et réalisé par :

ZERROUKI Sofiane

MERCHICHI Mohamed Zakaria

Encadreur/Co-Encadrant :

Mme GUERBAI Y.

Mr ABBOUD M.

Soutenu, le 03/07/2023, devant le jury composé de :

MESSAOUDI	Noureddine	MCA	UMBB	Président
MAHDI	Ismahan	MCB	UMBB	Examinateur
GUERBAI	Yasmine	MCA	UMBB	Encadrant

Année universitaire : 2022/2023

Dédicaces

A Dieu le tout puissante, source de toute connaissance, qui m'a donné La force et la volonté pour achever ce modeste travail.

A ma mère celle qui m'a donnée la vie, le symbole de tendresse et qui S'est sacrifiée pour mon bonheur et ma réussite.

A mon père, école de mon enfance, celui qui a été mon ombre durant Toutes mes années d'études, et qui a veillé tout au long de ma vie à M'encourager et à me donner l'aide.

Que Dieu les Garde et les Protège.

A ma chère sœur Zineb.

A mes grands-parents que Dieu les Protège.

A toute ma famille un par un.

Ames amis, lynda, imene, riham, thanina, rym, ilham, anass

Et chacun par son nom.

A toute ma promo avec qui j'ai partagé de très bons moments

A toutes ces personnes je dédie ce modeste travail en termes d'amour et de profonde gratitude.

Mohamed Zakaria

Dédicaces

A Dieu le tout puissante, source de toute connaissance, qui m'a donné la force et la volonté pour achever ce modeste travail.

A mon très cher père, pour ses encouragements, son soutien, surtout pour son amour et son sacrifice afin que rien n'entrave le déroulement de mes études

A ma mère, décédée il y a peu et qui serait contente d'apprendre que son fils a enfin terminé le travail qu'il avait commencé.

A ma 2éme mère, qui me donne toujours l'espoir de vivre et qui n'a jamais cessé de prier pour moi.

A mes chères sœur Samira, Chaïma et Hayat.

A mon cher petit frère Ishak.

A mes grands-parents que Dieu les Protège.

A toute ma famille un par un.

A mes amis, chacun par son nom. Spécialement groupe ferr

A une personne spéciale qui ne m'a jamais laissée

A toute ma promo avec qui j'ai partagé de très bons moments

A mon binôme Zaki pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

A toutes ces personnes je dédie ce modeste travail en termes d'amour et de profonde gratitude.

Sofiane

Remerciements

Le travail présenté, dans le cadre de ce Projet de Fin d'Etudes, a été développé et mis en œuvre à MOBILIS.

Louange à Allah, le tout Puissant, pour nous avoir donné la force, la santé et la capacité nécessaire pour accomplir ce modeste travail et pour nous avoir permis de le mener à bien.

On adresse nos remerciements les plus sincères à notre encadreur Mme GUERBAI YASMINE, Maître Conférences classe A à l'université, pour avoir accepté de codiriger ce travail et de suivre la rédaction de ce mémoire. On la remercie pour ses précieux conseils et pour son aide permanente. On a eu le privilège de travailler sous ses ailes. On a apprécié considérablement ses qualités et ses valeurs. Son sens du devoir et ses encouragements inlassables nous ont énormément marqué. On saisit cette occasion pour lui témoigner notre profonde gratitude.

On ne saurait jamais assez remercier notre Co-encadreur M. ABBOUD MOHAMED et KHENIEN RAOUF, Ingénieurs Réseaux à MOBILIS. Qu'ils trouvent ici l'expression de notre respectueuse considération et notre profonde gratitude pour toutes leur qualités scientifiques et humaines, pour leur aide précieuse et bénéfique, pour leur orientations constructives et riches. On tient à les remercier chaleureusement pour nous avoir fait confiance en nous confiant la réalisation de ce projet d'actualité et pour nous avoir réservé le meilleur accueil malgré leur obligations professionnelles.

Nos vifs remerciements vont aux membres du jury pour avoir accepté d'examiner et d'évaluer le travail réalisé dans ce mémoire. On tient à exprimer également nos respectueux remerciements à tous nos enseignants qui ont contribué à notre formation.

On remercie chaleureusement notre famille, nos parents, en premier lieu, pour leur soutien moral, leurs encouragements et leur patience durant les étapes difficiles de ce travail.

On gardera un très bon souvenir des moments passés à l'université en compagnie de nos amis dont notamment la promotion Master 2. On tient à les remercier vivement pour leur sympathie et leur gentillesse.

Enfin, on témoigne nos remerciements à tous ceux qui n'ont pas été cités mais qui ont contribué d'une manière ou d'une autre, chacun à sa manière, à l'élaboration et au bon déroulement de ce modeste travail.

Mohamed Zakaria

Sofiane

Chapitre I : Réseaux Internet et SDN

Chapitre II : Protocol BGP et application de défense "ARTEMIS"

Chapitre III : Mise en place de l'application "ARTEMIS"

A

AS : Autonomes System

API : Application Programming interface

ARTEMIS : Automatic and **R**eal-**T**ime DEtection and **MI**tigation **S**ystem

ASN : Advance Shipping Notice

B

BGP: Border Gateway Protocol

C

CAIDA : Center for Applied Internet Data Analysis

CLI : Command-Line Interface

E

EGP : External Gateway Protocol

EIGRP : Enhanced Interior Gateway Routing Protocol

E-BGP : External Border Gateway Protocol

F

FAI : *Fournisseur d'Accès à Internet*

I

IGP : Internal Gateway Protocol

IGRP : Interior Gateway Routing Protocol

I-BGP : Internal Border Gateway Protocol

N

NFV : Network function Virtualization

NAT : Network Address Translation

O

ONOS : Open Network Operating System

OSPF : Open Shortest Path First

ONF : Open Networking Fondation

ODL : Open Day Light

OSI : Open Systems Interconnection

OVS : Open *Virtual Switch*

P

PIP 3 : Package Installer for Python

R

R : Router

RIP : Routing Information Protocol

RIPE RIS : Réseaux IP Européens Routing Information Service

RRC : Remote Route Collector

RAM : *Random Access Memory*

S

SDN : *Software-Defined Networking*

SSL : *Secure Socket Layer*

T

TCP : Transmission Control Protocol

TLS : Transport Layer Security

X

XML : Extensible Markup Language

Introduction Générale

Information

« D'une image d'un objet, le cerveau humain retire une multitude d'informations, est-il possible de doter une machine d'une telle capacité d'analyse ? »

-Jean Serra-

Introduction générale

Internet a connu une croissance exponentielle, due en grande partie à la mise en œuvre de la technologie IP. Cependant, cette technologie présentait certains problèmes dans les années 90, tels que la congestion du réseau, des tables de routage trop volumineuses, la croissance et la complexité des réseaux, qui impactaient les performances des routeurs. De plus, le Border Gateway Protocol (BGP) manquait d'autorisation et avait une distribution naturelle, ce qui rendait possibles les incidents de détournement de préfixe. Par conséquent, il est essentiel de développer de nouvelles technologies capables de répondre aux changements, aux erreurs et aux charges pour résoudre ces problèmes.

La plupart des opérateurs ont pris la décision de migrer progressivement tous leurs réseaux vers des réseaux définis par logiciel (SDN). Cette approche donne la priorité à une réduction de la complexité et à la capacité d'adapter les réseaux aux préférences de leurs utilisateurs, de manière intuitive, rapide et fiable.

Dans les réseaux conventionnels, les routeurs utilisent des protocoles de routage distribués pour calculer l'interface de sortie appropriée pour diriger les paquets vers leurs destinations prévues. Les réseaux définis par logiciel (SDN), quant à eux, impliquent un ou plusieurs contrôleurs qui gèrent le calcul des routes, reléguant de fait les routeurs à de simples outils de transmission. Cette innovation permet le déploiement rapide de nouveaux services, tout en réduisant drastiquement les coûts.

La technologie du réseau défini par logiciel (SDN) est un outil puissant et inestimable à l'époque moderne. Néanmoins, en raison de l'absence d'autorisation au sein du réseau Internet, il existe un besoin pressant pour un filtrage de plus en plus rapide, rentable et précis des annonces d'itinéraire.

Pour faire face à ces contraintes, le groupe grec INSPIRE FORTH et le Center for Applied Internet Data Analysis (CAIDA) ont collaboré pour créer une solution viable pour les réseaux SDN qui peut se défendre contre les attaques de détournement de préfixe BGP. L'application, connue sous le nom de "ARTEMIS" ou Automatic and Real-Time Detection and Mitigation System, est conçue pour fournir aux opérateurs de réseau une multitude de fonctionnalités qui sont essentielles dans le paysage technologique en constante évolution d'aujourd'hui. Ces fonctionnalités incluent un degré élevé de précision, de vitesse, de flexibilité et de confidentialité.

L'application « ARTEMIS » a été spécialement conçu pour les administrateurs réseau. Sa fonction principale est de permettre l'identification instantanée et la résolution automatique des incidents impliquant le détournement de préfixe qui peuvent se produire en ce qui concerne les préfixes qui relèvent de leur domaine de contrôle administratif.

ATM Mobilis, filiale du groupe Algérie Télécom, est le premier opérateur de téléphonie mobile en Algérie. Elle contient une importante direction DoMR « Direction Opération et Maintenance Réseau » ou notre projet s'est déroulé. Cette direction a pour mission de fournir des services de gestion de réseau et assurer un bon fonctionnement de réseau.

Au cours de notre projet, nous explorerons l'importance d'un contrôleur SDN couramment utilisé dans divers contextes. Cependant, nous nous concentrons sur un scénario d'utilisation solitaire - en particulier, celui de "ARTEMIS" géré par le contrôleur ONOS (Open Network Operating System), qui utilise le protocole OpenFlow.

Notre solution a pour objectifs techniques de :

- Mettre en œuvre « ARTEMIS » en tant qu'application multi-module s'exécutant sur un Contrôleur ONOS.
- Détecter une attaque par détournement de préfixes en quelques secondes.
- Atténuer complètement le détournement en quelques minutes.
- Permettre aux AS légitimes de contrer rapidement le détournement en se basant sur les données qu'ils observent eux-mêmes sur le plan de contrôle.

Ce mémoire est organisé en trois chapitres présentés comme suit :

Le premier chapitre est dédié à la technologie SDN. Cette technologie a révolutionné les réseaux traditionnels et a permis l'apparition continue de nouvelles applications de gestion des réseaux. Donc, on évoquera, dans ce chapitre, le principe du SDN, son architecture et le protocole standard utilisé, à savoir, OpenFlow

Le deuxième chapitre se concentre sur l'introduction du protocole de routage BGP (Border Gateway Protocol), qui constitue le cœur du fonctionnement d'Internet. Ce protocole permet l'échange d'informations de routage et d'accessibilité des réseaux entre les systèmes autonomes (AS). Nous abordons également le problème fréquent du détournement de préfixe, qui engendre des perturbations de routage et des pertes économiques sur Internet. Ensuite, nous présentons l'application de défense contre les détournements de préfixe BGP, connue sous le nom de « ARTEMIS ».

Le troisième chapitre est dédié à la partie pratique de notre projet, où nous appliquons la technologie SDN avec le controlleur ONOS aux réseaux des opérateurs afin de réaliser une attaque de détournement de préfixe et de protéger les réseaux contre ce type d'attaque.

Notre travail se conclut par l'évocation de certaines perspectives qui pourraient constituer une extension de l'application "ARTEMIS". Parmi celles-ci, nous envisageons la prise en charge de plusieurs routeurs de fournisseurs, au-delà de Quagga, en utilisant des protocoles tels que NETCONF et YANG. De plus, nous envisageons d'utiliser la plateforme d'émulation GNS3 pour nos expérimentations. Ces développements potentiels ouvrent de nouvelles possibilités de recherche et d'amélioration de l'efficacité de notre solution.

Chapitre I

Réseaux Internet Et SDN

Savoir

Au fond, On ne sait que lorsqu'on sait peu,
Avec le savoir croît le doute
-Goethe-

I.1 Introduction

Au fil des années d'existence et de croissance des réseaux informatiques, l'augmentation considérable de la complexité des réseaux a entraîné des difficultés pour la gestion. Ce qui nécessite généralement leurs reconfigurations pour répondre aux changements, aux erreurs et aux charges

Une nouvelle perspective dans la technologie des réseaux, le réseau défini par logiciel (SDN) a retenu l'attention. Cette approche centralise le contrôle du réseau, permettant aux équipes d'exploitation du réseau de gérer un réseau entier comme une seule entité. Le contrôle des flux réseau est mis en œuvre à un niveau global d'abstraction via le protocole OpenFlow, plutôt qu'un contrôle individuel au niveau de l'équipement.

Dans ce chapitre, nous allons expliquer les différents plans fonctionnels des routeurs classiques, ainsi que la classification de routage dynamique. Nous expliquons ce qu'est le SDN et l'OpenFlow. Nous donnons les architectures de chacun et leurs fonctionnements.

I.2 Routeurs classiques

Un routeur est un équipement physique ou virtuel qui transmet des informations entre deux ou plusieurs réseaux informatiques à commutation par paquets. Un routeur inspecte l'adresse IP de destination d'un paquet donné, calcule le meilleur chemin pour qu'il s'atteigne sa destination, puis le transmet en conséquence.

Selon ses fonctionnalités, un routeur peut être deviser en deux plans :

I.2.1 Plan de contrôle

Le plan de contrôle est la partie du réseau qui contrôle la façon dont les paquets sont transmis, c'est-à-dire la façon dont les données sont envoyées d'un endroit à un autre. Par exemple, le processus de création de tables de routage est considéré comme faisant partie du plan de contrôle. Les routeurs utilisent divers protocoles pour identifier les chemins réseau et stocker ces chemins dans des tables de routage [1].

I.2.2 Plan de données

Contrairement au plan de contrôle, qui décide comment transférer les paquets, le plan de données transmet en fait les paquets. Le plan de données est également appelé plan de transmission.

Considérez le plan de contrôle comme des feux de circulation fonctionnant aux intersections de la ville. Pendant ce temps, le plan de données (ou plan de transfert) ressemble plus à une voiture conduisant sur la route, s'arrêtant à une intersection et obéissant à des feux de circulation [1].

I.3 Routage

Le routage sur Internet est l'opération par laquelle un élément (courrier, appels, paquets IP, etc.) va être acheminé d'un endroit à un autre. Cette opération est effectuée par des routeurs à grande capacité. On dispose deux types de routage :

I.3.1 Routage statique

Les informations sont mises à jour manuellement à chaque modification topologique de l'inter-réseau.

I.3.2 Routage dynamique

Cela consiste à trouver le chemin le plus court entre deux points du réseau en fonction de l'encombrement et de l'état du réseau. Les informations relatives à la route sont mises à jour automatiquement entre les routeurs [2].

I.4 Classification des protocoles de routage

Dans les protocoles de routage dynamiques, l'accessibilité du réseau de destination dépend de l'état du réseau.

I.4.1 Classification par mécanisme de travail et algorithme

Les protocoles de routage dynamique peuvent être classés en protocoles de routage vectoriel à distance et en protocoles de routage d'état de liaison selon leurs mécanismes de travail et algorithmes.

I.4.1.1 Protocoles de routage de vecteur de distance

Les algorithmes de routage utilisent un vecteur pour déterminer la direction du réseau et la distance par rapport aux liaisons individuelles. Cette dernière métrique est mesurée par le nombre total de sauts vers lesdits liens. Les nœuds routeurs échangent de manière autonome des entrées de table de routage entre eux à intervalles fixes, même en l'absence de modifications du réseau. Lorsqu'un routeur reçoit une mise à jour, il vérifie chaque chemin connu avant d'ajuster sa table de routage selon les besoins. Les protocoles d'acheminement des vecteurs de distance comprennent : BGP et EIGRP, etc [3].

I.4.1.2 Protocoles de routage d'état de liaison

Les protocoles de routage d'état de liaison utilisent l'algorithme d'état de liaison (LS). Au lieu de simplement apprendre les routes des routeurs voisins, les routeurs effectuant cet algorithme divisent les routeurs en zones, collectent des informations d'état de lien de tous les routeurs de la zone, et génèrent une topologie de réseau basée sur les informations d'état de lien, après quoi chaque routeur calcule les routes vers chaque réseau en fonction de la topologie.

Les protocoles de routage à l'état de liaison comprennent OSPF, IS-IS, etc [4].

I.4.2 Classification par aire de travail

Par zone de travail, les protocoles de routage dynamique peuvent être classés en protocoles de passerelle intérieure et en protocoles de passerelle extérieure.

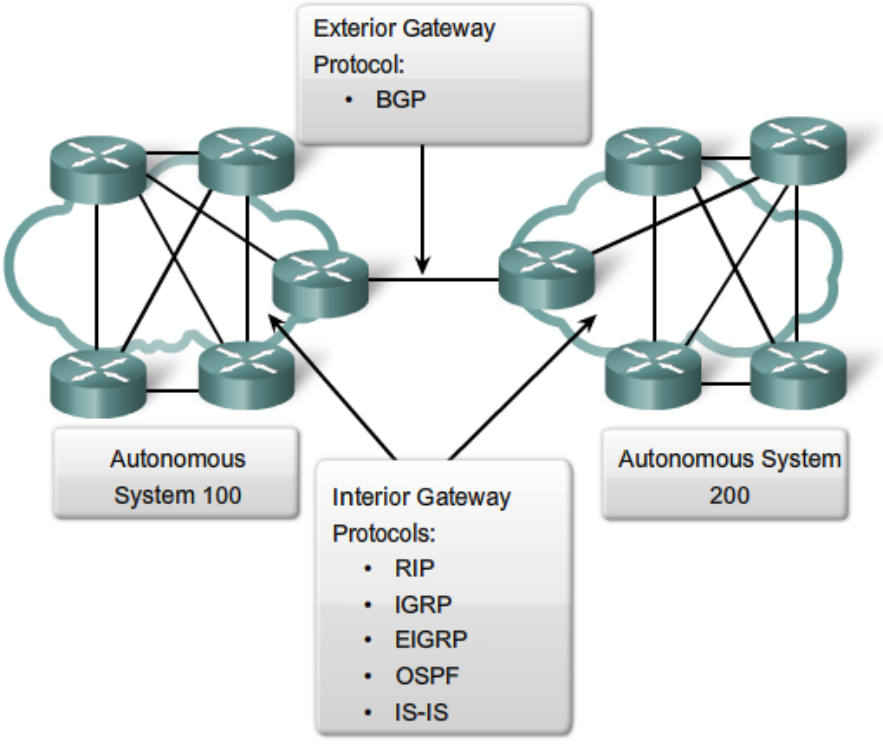

Figure I.1. *IGP et EGP*

I.4.2.1 Interior Gateway Protocol (IGP)

Interior Gateway Protocol (IGP) est un protocole de routage responsable du routage dans un système autonome. Il est chargé de s'assurer que chaque routeur dans un domaine suit la même manière de représenter les informations de routage et les mêmes règles pour la publication et le traitement des informations. Il est principalement utilisé pour découvrir et calculer des itinéraires.

Les protocoles de passerelle intérieure comprennent RIP, OSPF, EIGRP, IS-IS, etc [5].

I.4.2.2 Exterior Gateway Protocol (EGP)

Exterior gateway protocol (EGP) est conçu pour être utilisé entre des réseaux contrôlés par deux organisations différentes. Les EGP sont généralement utilisées entre les fournisseurs de services Internet (FSI) ou entre une entreprise et un FSI.

Un EGP doit isoler les systèmes autonomes. Parce que les systèmes autonomes sont gérés par différentes administrations, les réseaux doivent avoir un protocole pour communiquer entre différents systèmes [6].

I.5 SDN « Software Defined Networking »

I.5.1 Définition

Le Software Defined Networking (SDN) est un modèle d'architecture réseau qui permet aux administrateurs de gérer les services réseaux par abstraction des fonctionnalités.

Le SDN découple les fonctions de contrôle et de transfert des données afin d'avoir une infrastructure physique complètement exempte de tout service réseau. Dans ce modèle, les équipements réseau se contentent d'implémenter des règles dictées par les applications. Plus besoin de protocoles de routage sur ces équipements : un contrôleur central voit le réseau dans sa globalité et injecte directement les règles sur chaque équipement [7].

Le SDN rend les réseaux programmables via un contrôleur centralisé. Aujourd'hui les périphériques réseau prennent leurs propres décisions internes sur l'acheminement du trafic, en s'appuyant sur des protocoles distribués comme OSPF et BGP.

Le SDN établit une séparation claire entre le plan de contrôle (qui définit comment acheminer le trafic) et le plan de données (la partie des commutateurs et routeurs qui transfère effectivement les données).

Le contrôleur SDN perçoit l'état du réseau et des flux de données dans leur globalité. Il calcule et installe les chemins de données optimales dans le réseau sous son contrôle. Les équipements réseau (routeurs et commutateurs) se contentent d'appliquer les règles prescrites par le contrôleur [9].

I.5.2 Architecture de SDN

L'architecture SDN est aujourd'hui reconnue comme permettant d'ouvrir le réseau aux applications. Cela passe par :

- Autoriser les applications à programmer le réseau afin d'accélérer son déploiement.
- Permettre au réseau d'identifier les applications transportées pour mieux les gérer (qualité de service, sécurité, gestion du trafic...).

L'architecture SDN sépare le plan de contrôle du plan de transfert des données. Elle comprend trois couches :

- **La couche infrastructure :** Elle consiste en les commutateurs physiques du réseau. Elle contient les éléments réseau responsables du routage du trafic et qui supportent le protocole OpenFlow qu'ils partagent avec le contrôleur.
- **La couche de contrôle :** C'est une couche logicielle, basée sur le contrôleur SDN qui offre une vue globale du réseau et des équipements d'infrastructure. Le contrôleur réside sur le serveur et gère les règles et le flux de trafic sur le réseau [10].
- **La couche applicative :** Elle apporte l'automatisation des applications à travers du réseau, et à l'aide des interfaces programmables. Elle permet aux administrateurs de configurer, gérer, sécuriser et optimiser les ressources du réseau via des programmes.

Ainsi, dans une approche SDN, une grande partie du calcul associé au routage est retirée des routeurs. Le contrôleur est alors chargé du calcul et de la mise à jour des tables de routage [11].

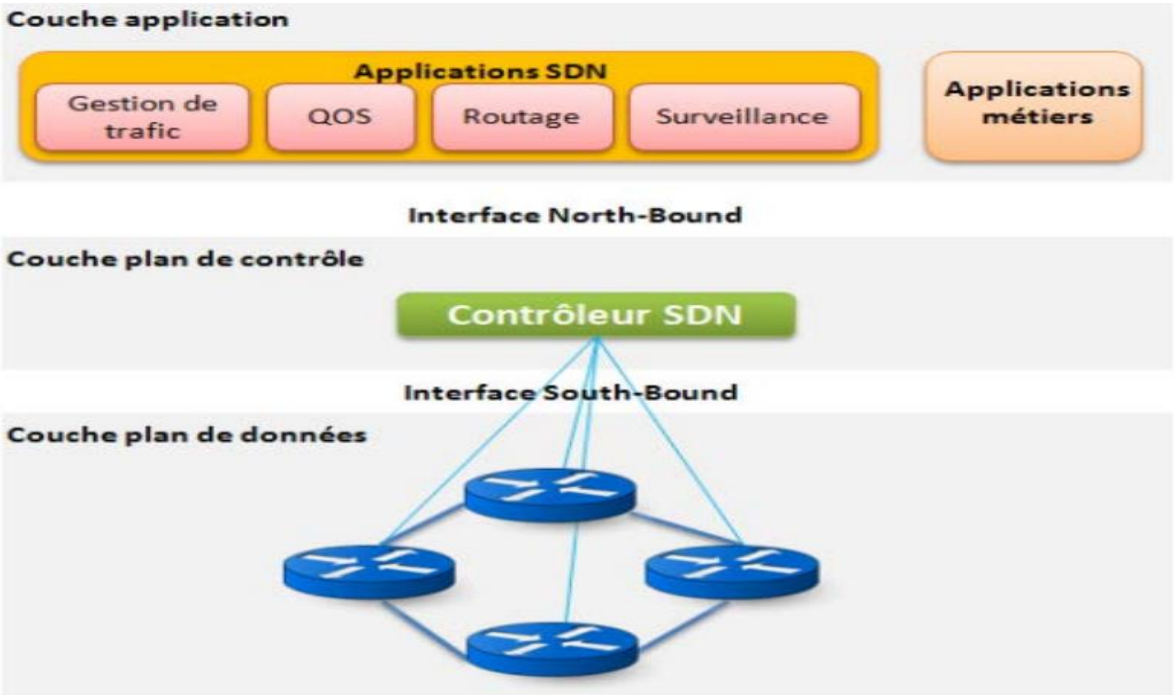

Figure I.2. *Architecture de SDN*

I.5.3 Différents modèles pour SDN

Le SDN recouvre toutes les solutions qui permettent une meilleure interaction entre la programmation réseau et les applications. Différentes solutions coexistent en fonction des besoins : les solutions de déploiement simplifié dans les data centers sont distinctes de celles permettant un meilleur contrôle de l'éclairage urbain par exemple.

On distingue plusieurs modèles de SDN :

- **La programmabilité individuelle :** les applications communiquent directement avec chaque équipement via des API. L'application commande directement chaque nœud du réseau.
- **La programmabilité via un contrôleur :** une application émet une commande globale à un contrôleur qui la décompose ensuite pour les équipements concernés. C'est le modèle le plus populaire car il simplifie la gestion du réseau.
- **La création d'un réseau virtuel au-dessus du réseau physique :** les applications créent leur propre réseau "overlay" indépendant du réseau sous-jacent. Ce dernier a alors uniquement un rôle de connectivité entre les nœuds, le réseau overlay assurant tous les services [7].

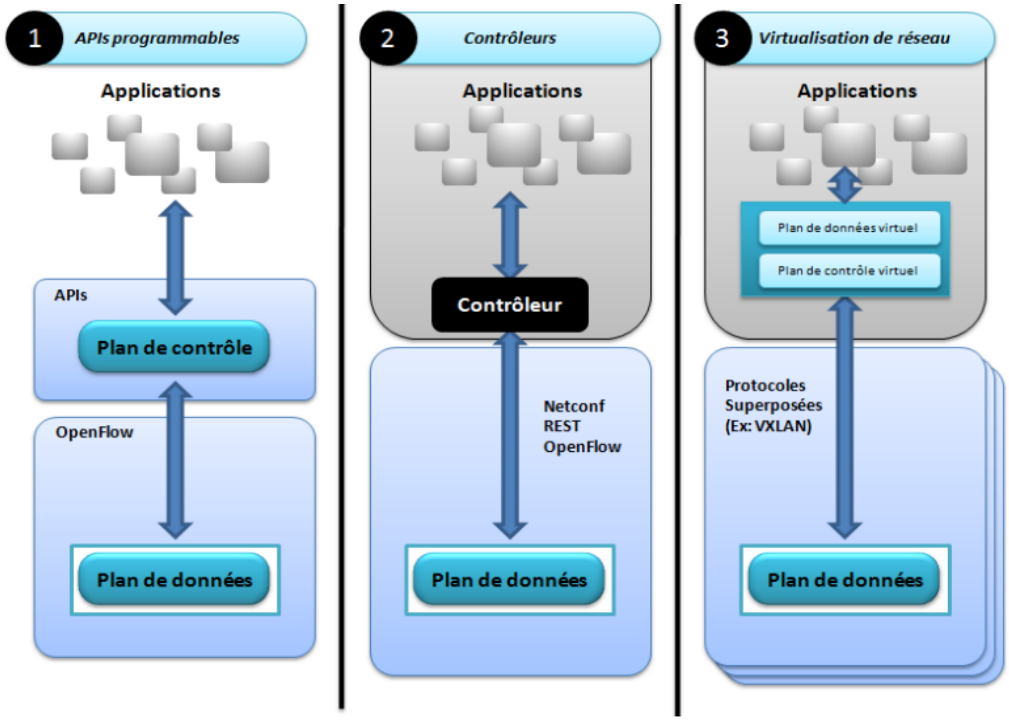

Figure I.3. *Les modèles du SDN*

I.5.4 Interfaces de communication

Le SDN utilise deux interfaces de communication :

❖ **L'interface sud (Southbound) :**

Permet au contrôleur d'interagir avec les nœuds de la couche d'infrastructure, tels que les commutateurs qui contiennent des tables de flux spéciales. Ces tables stockent et gèrent des règles appropriées à chaque flux plutôt que des adresses MAC.

❖ **L'interface nord (Northbound) :**

Sert à programmer les éléments de transmission en utilisant l'abstraction du réseau fournie par le plan de contrôle. Les interfaces nord sont considérées davantage comme des API que comme protocole de programmation et de gestion de réseau. Il n'y a pas de norme pour les interfaces nord et plusieurs niveaux d'abstraction et différents cas d'utilisation peuvent être caractérisés. Ainsi, il peut y avoir plusieurs interfaces nord pour servir tous les cas d'utilisation. Les industriels ont proposé une API basée sur REST API pour fournir une interface programmable utilisable par les applications du commerce [12].

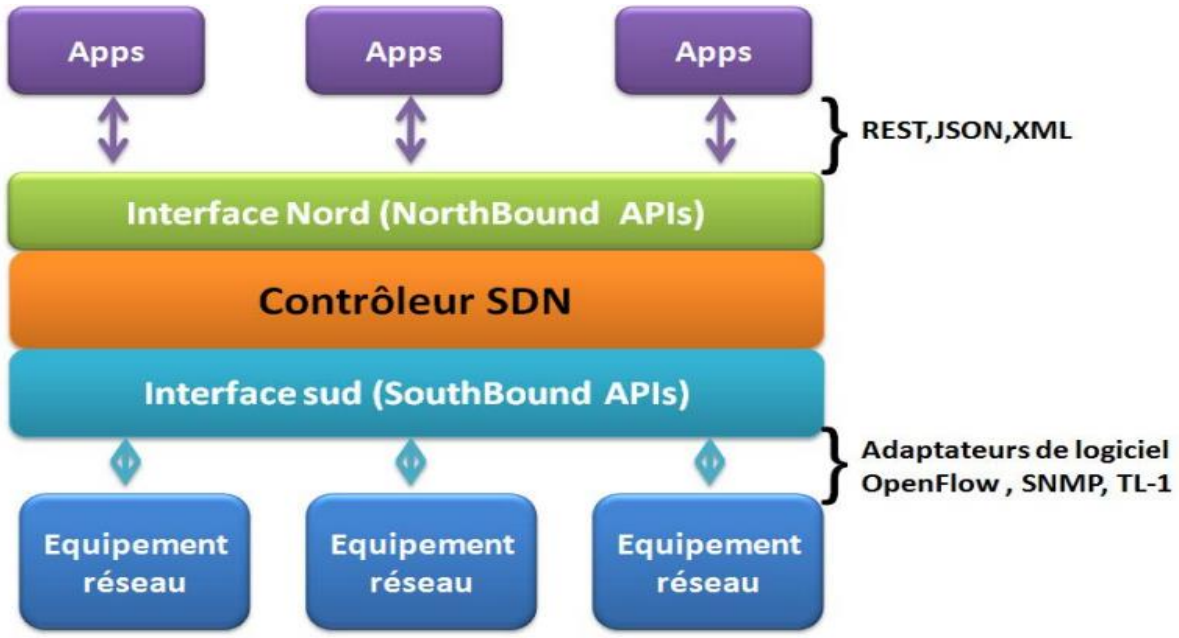

Figure I.4. *Interfaces sud et nord du SDN*

I.5.5 Applications SDN

Le SDN a de nombreuses applications dans les environnements réseaux, simplifiant le développement et le déploiement de nouveaux services et protocoles réseaux. Selon l'IDC, les équipements d'infrastructure, les logiciels de virtualisation, les applications de réseau et de sécurité sont évaluées à 12,5 milliards de dollars sur le marché en 2020 [13].

Le SDN est également bénéfique pour les centres de données et le cloud computing, où il peut optimiser la centralisation des données, accélérer le traitement et simplifier la gestion des commutateurs. Les applications multimédias comme la vidéo à la demande, la vidéoconférence et la WebTV nécessitent des ressources réseau stables et tolèrent les erreurs et les retards de transmission. Avec le SDN, des chemins différents pour les divers flux de trafic peuvent être sélectionnés, en utilisant une vue centralisée du réseau offerte par SDN.

En matière de sécurité, le SDN permet de détecter rapidement les attaques et de limiter leurs effets grâce à l'architecture centralisée. Des mécanismes ont été proposés pour détecter les attaques par déni de service (Dos et DDos). Le SDN est également utile pour les réseaux sans fil, où des nouvelles architectures basées sur la séparation du plan de contrôle et de données ont été proposées, tels que le SDWN (Software Defined Wireless Network), le CSDN (CellularSDN) et le SoftAir.

I.5.6 Avantages du SDN

SDN apporte de nombreux avantages dans la gestion des réseaux informatiques :

- **Flexibilité :** SDN offre une grande flexibilité dans la gestion du réseau. Il devient facile de rediriger le trafic, d'inspecter des flux particuliers, de tester de nouvelles stratégies ou de découvrir des flux inattendus.
- **Réduction des coûts d'exploitation :** Avec son contrôleur, SDN garantit une politique réseau unifiée et à jour. En effet, l'administrateur va simplement spécifier une nouvelle règle et le contrôleur adaptera la configuration pour envoyer des règles cohérentes dans chaque dispositif pertinent.
- **Routage simplifié :** SDN permet de gérer les informations de routage de manière centralisée en déléguant le routage et en utilisant une interface pour le contrôleur [14].
- **Gestion du Cloud facilitée :** SDN permet une gestion simple d'une plateforme Cloud. En effet, la dynamique apportée par SDN traite des problèmes spécifiques aux Clouds tels que l'évolutivité, l'adaptation, ou des mouvements de machines virtuelles.

- **Matériel moins coûteux :** SDN a tendance d'utiliser des technologies standard et de base pour contrôler les équipements du réseau, tandis que la puissance de calcul n'est requise qu'au niveau du contrôleur. Ainsi, les équipements de réseau deviendront des produits à bas prix offrant des interfaces standard.

Avec SDN, il est plus simple de modifier les stratégies réseau car il suffit de changer une politique de haut niveau et non de multiples règles dans divers équipements de réseau. De plus, la centralisation de la logique avec des connaissances globales et une grande puissance de calcul, simplifient le développement des fonctions plus sophistiquées. Cette aptitude à programmer le réseau est l'élément clé de SDN [13].

I.6 OpenFlow

I.6.1 Présentation d'OpenFlow

OpenFlow est une spécification de l'Open Networking Foundation (ONF) qui définit une infrastructure de redirection basée sur les flux et une interface applicative-programmatique normalisée. OpenFlow est un standard ouvert qui vous permet de contrôler le trafic et d'exécuter des protocoles expérimentaux dans un réseau existant en utilisant un contrôleur. Les composants OpenFlow consistent en un contrôleur, un commutateur OpenFlow, et le protocole OpenFlow. Le protocole OpenFlow est un protocole Layer 2 qui permet à un contrôleur OpenFlow d'accéder au plan de données d'un commutateur OpenFlow activé via une connexion SSL ou TCP/IP.

En utilisant OpenFlow, vous pouvez contrôler les chemins de trafic dans un réseau en créant, supprimant et modifiant les flux dans chaque périphérique le long d'un chemin. Les entrées de flux spécifient les conditions de correspondance par rapport auxquelles les paquets sont comparés, et un ensemble d'actions (OpenFlow v1.0) ou d'instructions (OpenFlow v1.3.1) qui sont appliquées aux paquets correspondants [15].

I.6.2 Fonctionnement d'OpenFlow

Le contrôleur contrôle et gère le commutateur via le protocole OpenFlow. Le contrôleur et le commutateur établissent un canal OpenFlow, par lequel ils échangent des informations. Si un commutateur établit des connexions OpenFlow avec plusieurs contrôleurs, les contrôleurs notifient le commutateur de leurs rôles par le canal OpenFlow, et les contrôleurs livrent la base d'informations de transfert ou les tables de flux au commutateur par le canal OpenFlow. Le

commutateur génère des entrées ARP pour la transmission de données selon la base d'informations de transmission ou le tableau de flux [16].

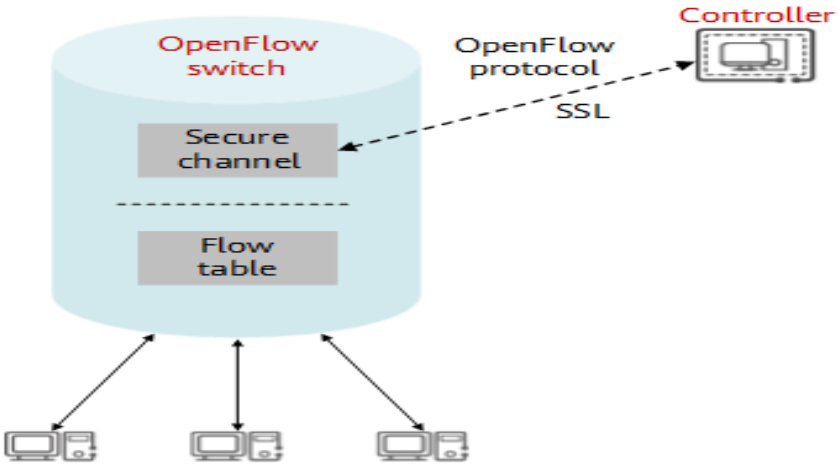

Figure I.5. *Architecture d'OpenFlow*

I.6.2.1 Contrôleur OpenFlow

Un contrôleur OpenFlow est le cerveau de l'architecture SDN et se trouve au niveau de la couche de contrôle pour instruire la transmission des données via le protocole OpenFlow. Actuellement, les contrôleurs OpenFlow classiques sont classés en deux types : les contrôleurs open-source et les contrôleurs commerciaux développés par les fournisseurs. Les contrôleurs open-source largement utilisés comprennent NOX, ONOS et ODL.

I.6.2.2 Canal sécurisé OpenFlow

Un canal sécurisé est établi entre un contrôleur et un commutateur OpenFlow. Par ce canal, le contrôleur contrôle et gère le commutateur, et reçoit la rétroaction du commutateur.

Les messages échangés sur le canal sécurisé OpenFlow doivent être conformes au format spécifié par le protocole OpenFlow. Le canal sécurisé OpenFlow est généralement chiffré à l'aide de Transport Layer Security (TLS), mais peut être exécuté directement sur TCP en texte clair dans OpenFlow 1.1 et versions ultérieures. Les messages OpenFlow suivants sont transmis sur le canal [17] :

- **Message Controller-to-Switch :** est envoyé par le contrôleur au commutateur OpenFlow pour gérer ou obtenir l'état du commutateur OpenFlow.
- **Message asynchrone :** est envoyé par le commutateur OpenFlow au contrôleur pour mettre à jour les événements réseau ou les changements d'état au contrôleur.

- **Message symétrique :** est envoyé sans sollicitation par le commutateur OpenFlow ou le contrôleur. Il est principalement utilisé pour établir une connexion et détecter si le pair est en ligne.

I.6.2.3 Commutateur OpenFlow

En tant que composant central du réseau OpenFlow, un commutateur OpenFlow est principalement responsable de la transmission à la couche de données. Il peut s'agir d'un commutateur/routeur physique ou virtualisé. Les commutateurs OpenFlow sont classés dans les types suivants en fonction de leur support pour OpenFlow :

- **Commutateur OpenFlow dédié :** est un périphérique OpenFlow standard qui ne prend en charge que la redirection OpenFlow. Le commutateur traite tout le trafic qui le traverse en mode OpenFlow, et ne peut pas effectuer de transfert de couche 2 ou 3 sur le trafic.
- **Commutateur compatible OpenFlow :** prend en charge la redirection OpenFlow et la redirection Layer 2/3. Il s'agit d'un commutateur commercial qui prend en charge les fonctionnalités OpenFlow telles que les tables de flux et les canaux sécurisés.
  Un commutateur OpenFlow redirige les paquets entrant dans le commutateur basé sur la table de flux, qui contient un ensemble d'entrées de politique indiquant au commutateur comment traiter le trafic. Les entrées de flux sont générées, maintenues et livrées par un contrôleur.

I.6.2.4 Flow table

Les messages de requête de fonctionnalité de table OpenFlow permettent à un contrôleur OpenFlow d'interroger les capacités des tables de flux existantes d'un périphérique géré par OpenFlow, ou de configurer ces tables pour correspondre à la configuration fournie.

Toutes les tables peuvent être configurées avec n'importe quel sous-ensemble des capacités de correspondance et d'action. Les tailles des tables peuvent également être modifiées lors de l'exécution. Lorsqu'une nouvelle configuration de table de flux est appliquée avec succès, les entrées de flux des anciennes tables de flux sont supprimées sans notification. Les tables de flux configurées dynamiquement ne sont pas persistantes pendant le redémarrage. Le pipeline par défaut apparaît lorsque l'appareil démarre.

Lors de la configuration d'un nouveau tableau de flux basé sur une demande du contrôleur OpenFlow, tout trafic en cours qui circule via les flux existants est supprimé [18].

I.6.3 Avantages d'OpenFlow

OpenFlow offre plusieurs avantages importants [19] :

- Performance et coût : En supprimant la charge de contrôle-traitement des commutateurs, OpenFlow permet aux commutateurs de se concentrer sur la circulation aussi rapidement que possible.
- En virtualisant la gestion de réseau, OpenFlow permet de construire et d'exécuter des réseaux moins coûteux.
- Implémentation et test de nouvelles fonctionnalités : OpenFlow permet aux administrateurs de déployer de nouvelles fonctionnalités en utilisant l'architecture réseau existante en les ajoutant dans le logiciel OpenFlow.
- Les fonctionnalités fonctionneraient alors sur plusieurs plateformes : les utilisateurs n'auraient pas à les implémenter dans le matériel ou le firmware de chaque fournisseur de commutateur.
- Grâce à son API ouverte, OpenFlow permet également aux administrateurs et aux chercheurs d'écrire leur propre logiciel de contrôle et d'essayer de nouvelles fonctionnalités de commutation importantes à taux plein.

Cela a été difficile dans le passé parce que les routeurs et les commutateurs des principaux fournisseurs manquaient d'API communes.

I.7 Conclusion

Dans ce chapitre, nous avons introduit les différents plans de routeur ainsi que des généralités sur le routage statique et dynamique. Nous avons présenté la technologie SDN avec son architecture et fonctionnement ainsi que le protocole OpenFlow.

Chapitre II

Protocole BGP et Application de Défense "ARTEMIS "

Technologie

La Technologie peut être utilisée pour le

Meilleur ou le pire. Elle a transformé

Notre manière de vivre

-Hugh Montefiore-

II.1 Introduction

Internet est composé de nombreux systèmes autonomes (AS) qui utilisent le protocole BGP (Border Gateway Protocol) pour transférer le trafic inter-domaines. Le manque de contrôle centralisé et de dispersion naturelle de BGP permet à un AS d'annoncer des routes ou des préfixes frauduleux appartenant à d'autres AS, détournant ainsi leurs préfixes.

Dans ce chapitre, nous présenterons une application multi-module nommée « ARTEMIS », fonctionnant sur ONOS, qui permet une détection et une atténuation rapides d'une attaque de détournement en quelques minutes.

II.2 BGP « Border Gateway Protocol »

Le Border Gateway Protocol est un protocole de communication. Il se charge dynamiquement de calculer les meilleurs chemins vers une destination et de les propager à travers les réseaux IP [20].

C'est un protocole de routage complexe qui est au cœur du fonctionnement d'Internet. Son objectif principal est d'échanger des informations de routage et d'assurer l'accessibilité de réseaux, appelés préfixes, entre les AS. Comme il circule sur TCP, il est considéré comme appartenant à la couche application du modèle OSI [21].

Figure II.1. *Border Gateway Protocol*

II.3 Caractéristiques du BGP

Les principales caractéristiques du BGP se résument comme suit [22] :

- Standard (interopérable) : compris et adopté par une vaste communauté de constructeurs et d'utilisateurs. Généralement, un protocole est dit standard lorsqu'il est considéré meilleur que les autres sur les plans technique et commercial.
- BGP est plus approprié lorsqu'au moins l'une des conditions suivantes existe :

- Un AS a plusieurs connexions avec d'autres AS.
- Un AS permet le routage de sélection politique et l'itinéraire pour le trafic entrant et sortant.
- Un AS permet aux paquets de traverser pour atteindre d'autres AS.

- BGP est un protocole de type Path Vector, qui est une évolution du type vecteur de distance. Il présente les améliorations suivantes :
  - Les mises à jour des chemins sont fiables et utilisent le protocole TCP.
  - Les mises à jour sont effectuées de manière incrémentielle et déclenchées uniquement lorsque nécessaire.
  - Des messages de maintien de la connexion (Keepalive) sont envoyés périodiquement pour vérifier la connectivité TCP.
  - Il est conçu pour être utilisé dans de vastes réseaux d'interconnexion.
- La convergence dans un réseau BGP est généralement lente, ce qui signifie que les changements de routage prennent du temps à se propager et à se stabiliser.
- Les mises à jour de routage sont envoyées uniquement lorsqu'il y a un changement, et seuls les éléments modifiés sont transmis, ce qui permet d'économiser de la bande passante.

II.4 Messages du protocole BGP

Une fois une connexion est ouverte, BGP échange plusieurs messages avec les paramètres de la connexion et les informations de routage, tous ces messages BGP sont diffuser vers un seul partenaire [23].

Les messages utilisés par BGP sont les suivants :

- **Message OPEN :** Ce message permet de démarrer une relation du voisinage entre deux BGP peer. Il contient une version du protocole BGP utilisé, un ID de routeur, un numéro AS et des Timer.
- **Message KEEPALIVE :** Ce message est envoyé périodiquement pour maintenir la relation du voisinage.
- **Message UPDATE :** Ce message permet d'annoncer des nouvelles routes et les chemins d'accès. Chaque route est caractérisée par son origine et son AS path.
- **Message NOTIFICATION :** Ce message est utilisé pour mettre fin à une session BGP en cas d'erreur.

- **Message ROUTE-REFRESH :** La capacité de rafraîchissement des routes est négociée dans le message OPEN. Ce message permet de demander la réannonce de certains préfixes après une modification de la politique de filtrage [23].

II.5 Relation de voisinage BGP

Pour établir une relation de voisinage BGP entre deux routeurs, il est nécessaire de configurer manuellement cette relation sur les deux routeurs. Contrairement à d'autres protocoles où les voisins sont détectés automatiquement, avec BGP, il est nécessaire de spécifier manuellement l'adresse du voisin afin d'établir la connexion [24].

II.5.1 BGP Peer

Un BGP peer, également connu sous le nom BGP Neighbors, est un terme spécifique utilisé pour des haut-parleurs BGP qui ont établi une relation de voisin. Les deux routeurs qui ont formé une connexion TCP pour échanger des informations de routage BGP sont appelés BGP peer ou voisins BGP (BGP Neighbors) [22].

Il existe deux types de relation BGP :

1. **External BGP**

Lorsque BGP tourne entre voisins qui appartiennent à différents AS, on l'appelle EBGP. Par défaut, les voisins EBGP doivent être connectés directement.

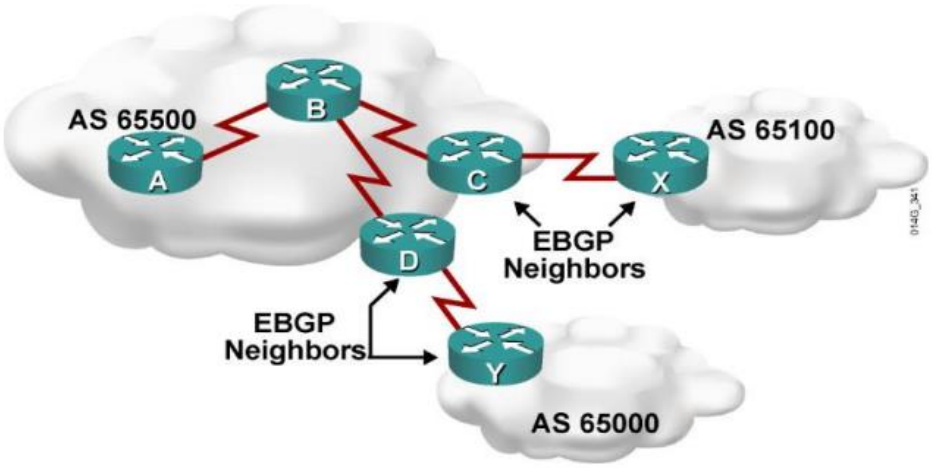

Figure II.2. *eBGP Neighbors*

2. **Internal BGP**

Lorsque BGP tourne entre voisins dans le même AS, on l'appelle IBGP. Les voisins n'ont pas à être directement connectés.

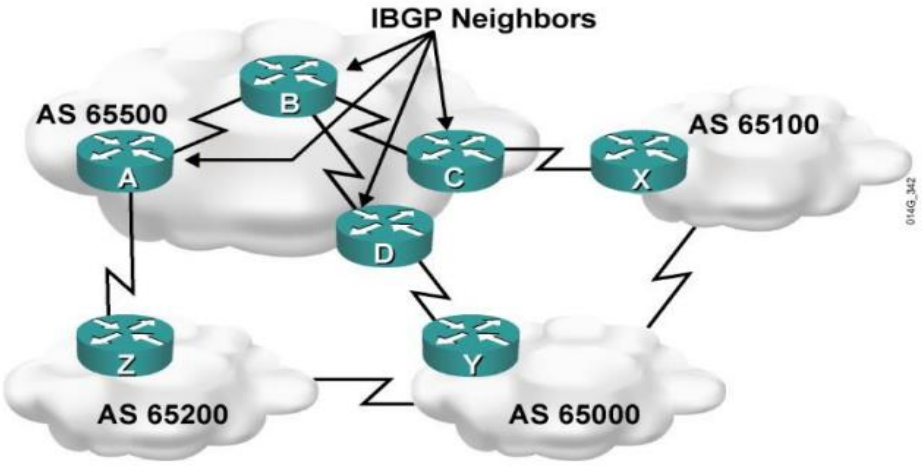

Figure II.3. *iBGP Neighbors*

II.5.2 Etapes d'une relation du voisinage

La figure illustre les étapes d'une relation du voisinage :

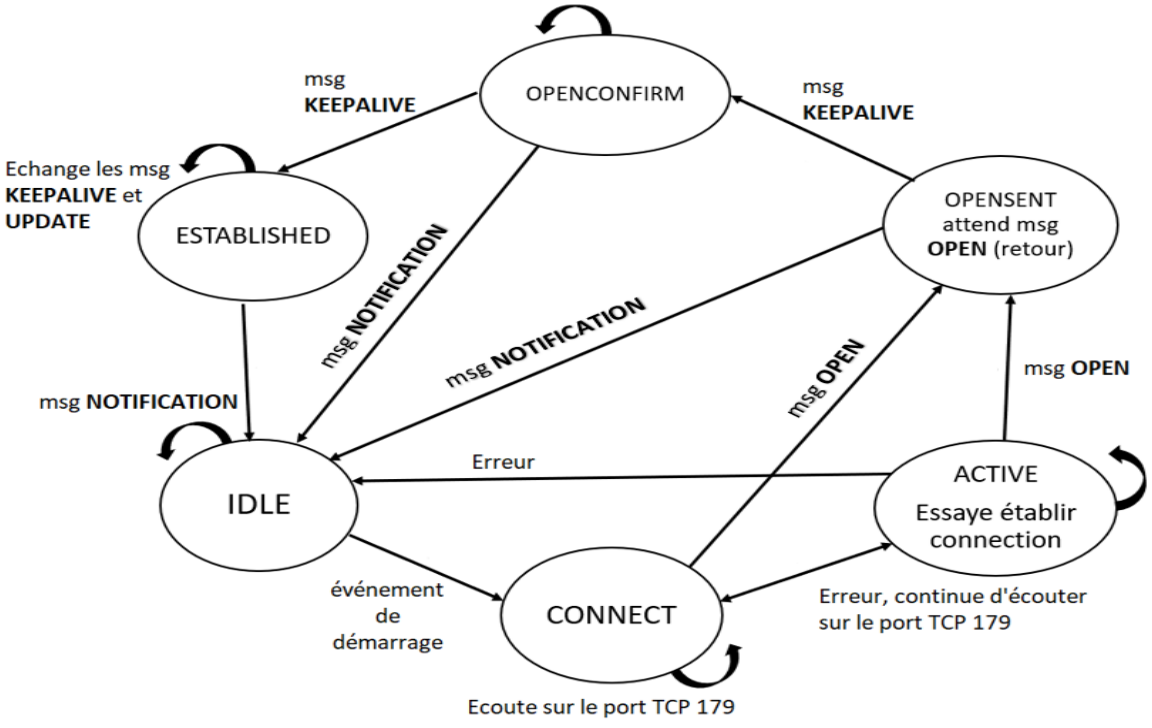

Figure II.4. *Étapes d'une relation du voisinage*

Ces étapes sont définies comme suit :

- **État Idle :** Premier état d'une session BGP. Quand une ouverture de session est demandée (configuration, démarrage du routeur), le routeur initie une session TCP, puis passe en mode « Connect » pour attendre la réponse.
- **État Connect :** En cas d'erreur, la connexion est interrompue et le processus revient à l'état Idle. Il attend que la connexion TCP soit établie, puis envoie le message OPEN et passe à l'état OpenSent. En cas d'erreur, il attend un délai prédéfini, continue d'écouter sur le port 179 et passe à l'état Active.
- **État Active :** La tentative de connexion avec le voisin a échoué (Timeout). Le routeur fait une nouvelle tentative. En cas d'échec, il retombe en mode Idle. En cas de succès, on passe en mode OpenSent. Ce statut n'est donc atteint qu'en cas d'échec en mode Connect.
- **État OpenSent :** Le message OPEN a été envoyé et BGP attend le message OPEN en retour. En l'absence d'erreur, BGP envoie un KEEPALIVE et passe à l'état OpenConfirm. Dans tous les autres cas, BGP envoie un message NOTIFICATION et retourne à l'état Idle.
- **État OpenConfirm :** Il attend un message KEEPALIVE et passe ensuite à l'état Established. Si un message NOTIFICATION est reçu, BGP revient à l'état Idle.
- **État Established :** La relation de voisinage est établie. Les routeurs peuvent s'échanger des routes.

II.6 Fonctionnement du protocole BGP

Le fonctionnement du protocole BGP s'articule de la manière suivante :

- Les connexions eBGP sont établies soit sur des liaisons point-à-point, soit sur des réseaux locaux. Pour garantir l'intégrité des paquets de la session BGP, la durée de vie (Time to Live, TTL) de ces paquets est fixée à 1. En cas de rupture de la liaison physique, la session eBGP est également rompue, ce qui entraîne la suppression de tous les préfixes appris par cette session et leur retrait de la table de routage.
- En revanche, les connexions iBGP sont généralement établies entre des adresses logiques qui ne sont pas liées à une interface physique spécifique. Cette configuration permet de maintenir la session iBGP active même en cas de rupture d'un lien physique, à condition qu'une alternative existe et qu'un protocole de routage interne dynamique tel que OSPF soit utilisé.

- Une fois la connexion établie entre deux routeurs, ils procèdent à l'échange d'informations sur les réseaux qu'ils connaissent et sur lesquels ils peuvent acheminer du trafic. Ils échangent également divers attributs associés à ces réseaux, tels que le chemin AS (AS Path), afin d'éviter les boucles et de sélectionner avec précision la meilleure route.

Le BGP choisit le meilleur chemin à l'aide des attributs qui sont classés selon six types :

- **Well-Known :** Ils sont standard et supportés par tous les constructeurs.
- **Optional :** Ils sont connus par certaines implémentations mais nécessairement reconnus par tous les routeurs BGP.
- **Mandatory :** Ils sont inclus dans les MAJ du routage BGP.
- **Discretionary :** Par défaut, ils ne sont pas inclus dans les MAJ du routage BGP.
- **Transitive :** Ils passent d'un AS à un autre.
- **Non-Transitive :** Il se caractérise par le contraire de Transitive.

II.7 Les Attributs de BGP

Lorsqu'un routeur BGP reçoit des mises à jour provenant de plusieurs systèmes autonomes décrivant différents chemins vers une même destination, il doit sélectionner un itinéraire unique et optimal pour atteindre cette destination, puis le transmettre à ses voisins. Cette décision est basée sur la valeur des attributs contenus dans la mise à jour, ainsi que sur d'autres facteurs configurables par BGP. Les attributs comprennent les éléments suivants [24] :

- **AS-path (attribut obligatoire) :** À chaque fois qu'une mise à jour traverse un système autonome, BGP ajoute son numéro d'AS à la mise à jour. L'attribut AS-path représente la liste des numéros d'AS que la mise à jour a traversés pour atteindre la destination. Cette liste s'allonge à mesure que la mise à jour se propage, évitant ainsi les redondances et les boucles.
- **Origin (attribut obligatoire) :** Cet attribut fournit des informations sur l'origine de la route et peut prendre l'une des valeurs suivantes :
  - IGP (route provenant du même système autonome que l'annonceur).
  - EGP (route apprise via le protocole de routage externe EGP) ou Incomplète (route apprise d'une autre manière, peut-être configurée statiquement).

- **Next hop (attribut obligatoire) :** Cet attribut contient l'adresse IP du routeur vers lequel l'information doit être envoyée pour atteindre le réseau. En général, il s'agit du routeur qui a émis la mise à jour BGP.
- **Weight (attribut optionnel) :** C'est un attribut spécifique à Cisco utilisé dans le processus de sélection de chemin lorsque plusieurs routes vers la même destination existent. Le poids est local au routeur sur lequel il est défini et n'est pas propagé dans les mises à jour de routage. Les routes ayant un poids plus élevé sont préférées lorsqu'il y a plusieurs chemins vers la même destination [20].
- **Local préférence (attribut optionnel) :** Cet attribut de préférence locale joue un rôle similaire à l'attribut de poids, mais il fait partie des informations de mise à jour de routage échangées entre les routeurs d'un même système autonome.
- **Multi-Exit Discriminator (MED) (attribut optionnel) :** Cet attribut fournit une indication aux routeurs voisins externes sur le chemin à privilégier vers un système autonome lorsque celui-ci possède plusieurs points d'entrée via différents routeurs externes de l'autre système autonome. Plus la valeur de MED est petite, plus le chemin est préféré. Cet attribut est échangé entre les systèmes autonomes, mais il n'est pas propagé au-delà d'un système autonome.
- **Community (attribut optionnel) :** L'attribut de communauté associé à une route est utilisé pour regrouper des destinations auxquelles des décisions de routage peuvent être appliquées.

II.8 Risques des interconnexions BGP

Plusieurs types de risques peuvent conduire à des problèmes de routage et des pertes économiques. Parmi ces risques, nous citons le détournement de préfixe BGP qui implique l'intrusion dans une session BGP en cours, c'est-à-dire que l'attaquant se fait passer pour l'un des voisins d'une session BGP.

II.8.1 Détournement de préfixe BGP

Le détournement de préfixe est l'acte d'absorber (une partie) du trafic destiné à un autre AS par la propagation de routes BGP erronées, ce qui est possible en raison de la confiance mutuelle implicite entre les routeurs BGP. Il peut être le résultat d'erreurs de configuration des routeurs ou d'intentions malveillantes.

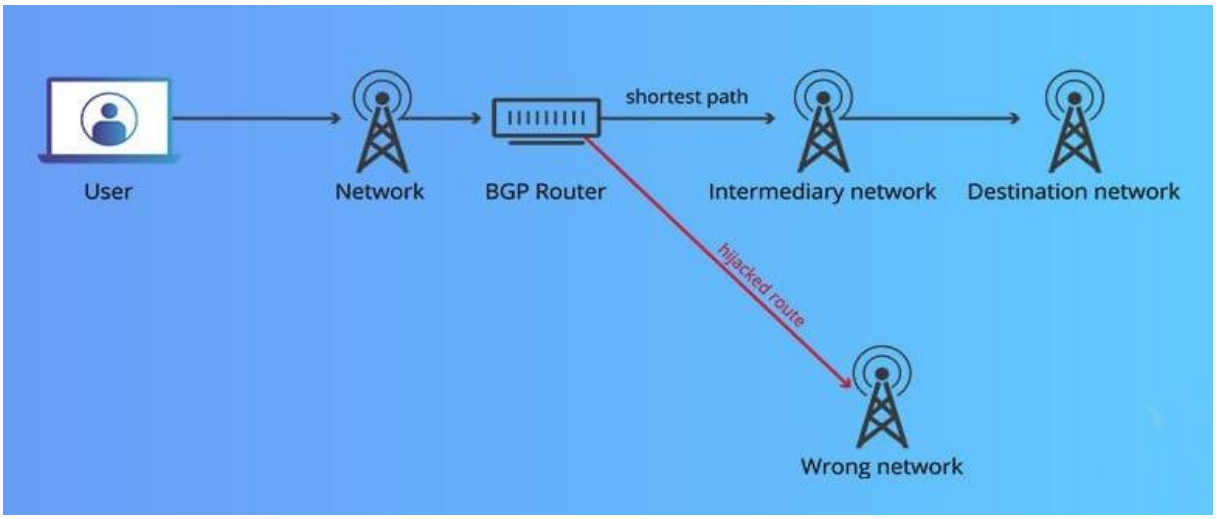

Figure II.5. *Réacheminent malicieux du trafic Internet*

II.8.2 Procédure de détournement de préfixe BGP

L'annonce de route par inadvertance par un AS pour les préfixes IP en dehors de sa compétence peut proliférer sans contrôle et s'assimiler dans les tables de routage des routeurs BGP sur Internet. Le résultat : le trafic vers ces adresses IP peut être dirigé par inadvertance vers cet AS jusqu'à ce que les routes erronées soient détectées et rectifiées.

BGP donne la priorité à la route la plus brève et la plus détaillée vers l'adresse IP prévue. Afin de réaliser le piratage BGP, l'annonce de route doit [25] :

- Offrir un itinéraire plus spécifique en annonçant une plage d'adresses IP plus petite que les autres AS avaient annoncée précédemment.
- Pour faciliter un détournement de BGP, il faut soit être un opérateur AS, soit un acteur malveillant ayant infiltré un AS, dans le but de proposer une route plus courte vers des adresses IP spécifiques.

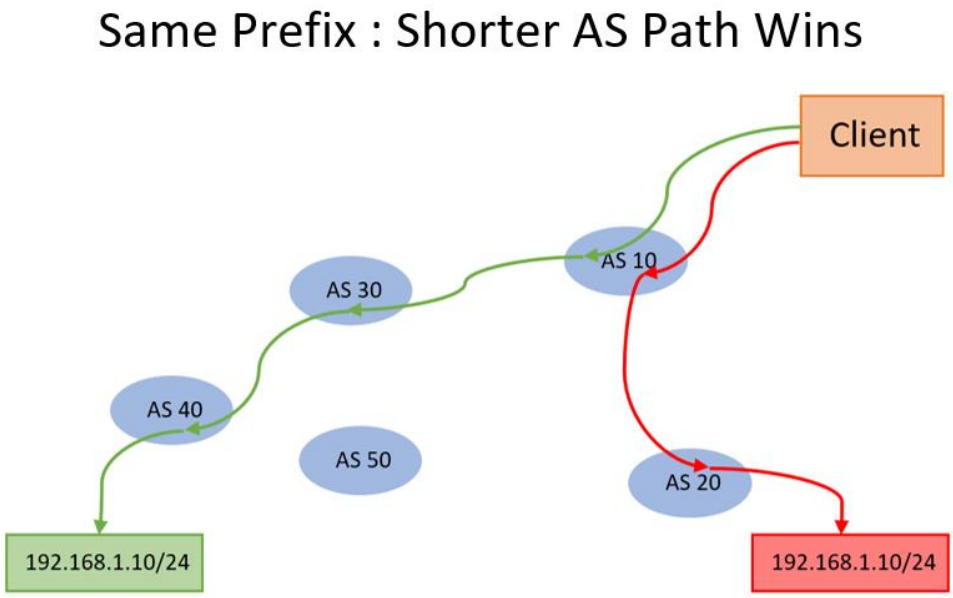

Figure II.6. *Détournement de préfixe BGP*

Les subtilités du détournement de BGP peuvent rendre son identification difficile, car des individus malveillants peuvent masquer leur activité sous différents AS. Alternativement, ils pourraient également promouvoir des préfixes IP inutilisés qui pourraient facilement passer.

II.8.3 Cas réels d'incidents de détournement de préfixe BGP

Dans cette section, nous citons quelques exemples des incidents de détournements de préfixes BGP :

- Le 22 janvier 2006, la société américaine Con Edison a annoncé des préfixes à la place de ses clients et d'autres AS sans relation avec eux. Le trafic des AS touchés a été redirigé vers le réseau de Con Edison.

- En 2008, la société publique pakistanaise Pakistan Telecom a tenté de censurer Youtube au Pakistan en mettant à jour le routage BGP de son site Web. Apparemment par accident, la nouvelle route a été annoncée au fournisseur en amont de Pakistan Telecom, et à partir de là, elle a été diffusée sur l'ensemble de l'Internet. Tout d'un coup, toutes les requêtes Youtube ont été dirigées vers Pakistan Telecom, ce qui a entraîné l'arrêt du site pendant des heures.

- En avril 2011, quelques blocs d'adresses IP appartenant à la société de télécommunications russe LinkTelecom ont été détournés pendant cinq mois et les adresses IP volées utilisées pour effectuer des activités malveillantes activités, comme

l'envoi de spam, l'hébergement de services web, la recherche de vulnérabilités sur les hôtes distants et la génération de trafic IRC [26].

- En avril 2018, un fournisseur russe a publié des préfixes IP (groupes d'adresses IP) qui appartenaient en réalité aux serveurs DNS Route53 d'Amazon. En bref, le résultat final est que les utilisateurs qui tentent de se connecter à un site Web de crypto-monnaie sont redirigés vers une fausse version du site contrôlée par des pirates. Les pirates ont pu voler environ 152 000 $ en crypto-monnaie [27].

II.9 ARTEMIS

ARTEMIS, Automatic and Real-Time dEtection and MItigation System, est le résultat des recherches entre le groupe INSPIRE, FORTH et le Center for Applied Internet Data Analysis (CAIDA), de l'University of California San Diego, États-Unis [28].

II.9.1 Définition

ARTEMIS est un système de défense contre le détournement de préfixe BGP, qui est basé sur une détection complète, précise et rapide opérée par l'AS lui-même, et permet une atténuation flexible et rapide des événements de détournement [29].

ARTEMIS utilise une surveillance en temps réel des données BGP (par exemple, les mises à jour BGP exportées par les collecteurs d'itinéraires) et peut détecter une attaque de détournement de préfixe en quelques secondes, et atténuer complètement le détournement en quelques minutes (par exemple, 2-5 minutes dans les expériences initiales sur Internet réel avec le banc d'essai PEERING) [30].

II.9.2 Architecture

Le système est composé de trois services principaux : surveillance, détection et atténuation, comme le montre la figure.

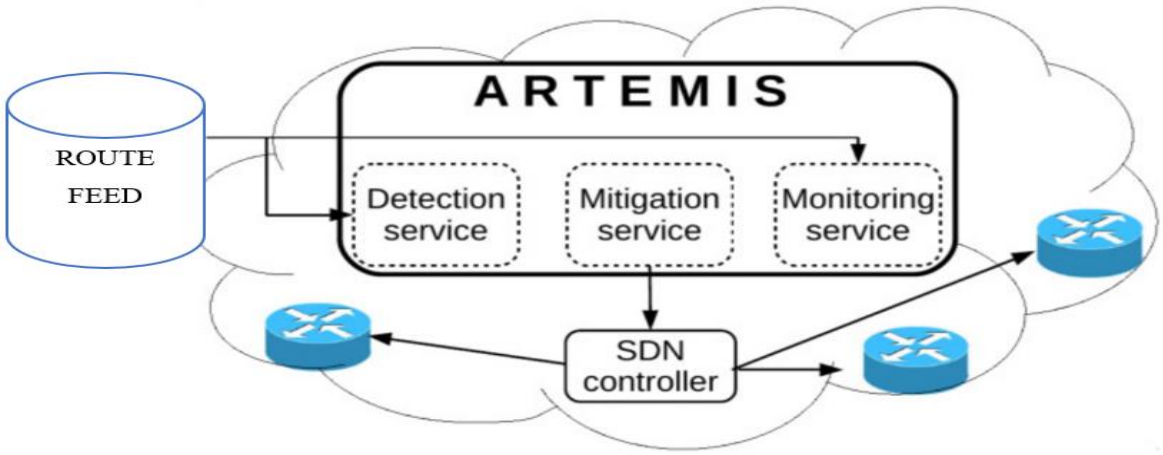

Figure II.7. *Architecture ARTEMIS*

II.9.2.1 Service de surveillance

Ce service fonctionne en continu et fournit des informations sur le plan de contrôle à partir de l'AS lui-même. Des services de streaming [31] :

- RIPE RIS collecte et stocke des données de routage Internet à partir de plusieurs endroits à travers le monde, en utilisant le RIPE NCC Routing Information Service (RIS). Les données RIS sont collectées par un certain nombre de collecteurs de routes RIS (RRC). Les données brutes sont également disponibles.
- BGPmon est un outil de nouvelle génération qui surveille les informations de routage BGP en temps réel. BGPmon rassemble les données BGP peer connectés et fournit des informations de routage en temps réel dans divers formats, comme XML.
- Périscope renvoie des mises à jour BGP en temps réel pour une liste donnée de préfixes et d'ASN.
- BGPstream est un logiciel Open Source pour l'analyse des données BGP en direct et historiques, soutenant la recherche scientifique, la surveillance opérationnelle et l'analyse post-événement.

II.9.2.2 Service de détection

Le service de détection fonctionne sans interruption et consolide les données du plan de contrôle de Périscope, du service de streaming RIPE RIS et de BGPmon. Alors que cette dernière renvoie les routes/mises à jour BGP pour les préfixes et ASN spécifiés, les deux premières sources offrent des informations en temps quasi réel. Avec cette approche multi-sources, la phase de détection connaît le délai le plus faible parmi toutes ses sources [32].

II.9.2.3 Service de mitigation

Dès qu'un détournement de préfixe est détecté, ARTEMIS entre en action et lance son service de mitigation. Dans le monde complexe du routage Internet, le préfixe le plus spécifique prime invariablement. Par conséquent, ARTEMIS modifie la configuration des routeurs BGP, en veillant à ce que les sous-préfixes du préfixe détourné soient annoncés avec précision. Une fois que le BGP a convergé, l'attaque de piratage est réprimée avec succès, permettant au trafic de reprendre son flux normal dans l'AS sécurisé par ARTEMIS.

Par conséquent, ARTEMIS suppose des autorisations d'écriture sur les routeurs du réseau afin de pouvoir modifier leur configuration BGP et atténuer l'attaque. Cela peut être accompli efficacement en exécutant ARTEMIS comme un ensemble de modules de niveau application, sur un contrôleur de réseau qui prend en charge le BGP, tel que l'ONOS [33].

II.10 Conclusion

Dans ce chapitre, nous avons présenté un aperçu du protocole BGP, en soulignant ses principales caractéristiques et fonctionnalités. De plus, nous avons cité les cas d'attaques de détournement de préfixe, qui sont connues pour causer des complications de routage et des dommages financiers sur Internet.

Par la suite, l'application de défense ARTEMIS a été introduite comme contre-mesure contre les attaques de détournement de préfixe BGP.

ARTEMIS fournit aux opérateurs de réseau une gamme de fonctionnalités souhaitables, notamment la précision, la vitesse, la confidentialité et la flexibilité.

Notre prochain chapitre présentera une application qui vise à annuler le détournement de préfixe en quelques minutes grâce à l'utilisation de l'approche ARTEMIS.

Chapitre III

Conception et Réalisation de l'Application « ARTEMIS »

Projet

Tu peux tout accomplir dans la vie si tu as
Le courage de le rêver, l'intelligence d'en
Faire un projet réaliste et la volonté de
Voir ce projet mené à bien

-Sidney A. Friedman-

III.1 Introduction

ARTEMIS est l'une des principales applications ONOS, dédiée aux administrateurs réseau. Elle permet de détecter et d'atténuer automatiquement, en temps réel, les incidents de détournement de préfixe contre les préfixes sous leur contrôle administratif.

Ce chapitre est consacré à la présentation de notre application qui a comme objectif de mettre en œuvre ARTEMIS en tant qu'application multi-modules fonctionnant sur ONOS. Nous débuterons par une démonstration de la topologie utilisée. Par la suite, nous détaillerons toutes les étapes que nous avons suivies lors de l'installation. Nous conclurons ce chapitre par les tests et les résultats obtenus.

III.2 ONOS « Open Network Operating System »

III.2.1 Définition

Le système d'exploitation des réseaux ouverts (ONOS) est le principal contrôleur SDN à source ouverte utilisé pour la création de solutions SDN/NFV (Network Function Virtualization) de nouvelle génération.

ONOS est le premier système d'exploitation de réseau SDN à source ouverte conçu pour les fournisseurs de services et les réseaux critiques. ONOS est conçu pour répondre aux exigences de haute disponibilité (HA), d'extensibilité et de performance de ces réseaux. ONOS aidera les fournisseurs de services à migrer leurs réseaux existants vers des boîtes blanches et apportera des caractéristiques de qualité opérateur (échelle, disponibilité et performance) au plan de contrôle du SDN. Réduire les coûts d'exploitation et d'investissement des fournisseurs de services.

L'architecture ONOS est conçue spécifiquement pour les exigences des réseaux de classe opérateur en matière de performance, de haute disponibilité et d'échelle, avec des abstractions bien définies aux figures suivantes :

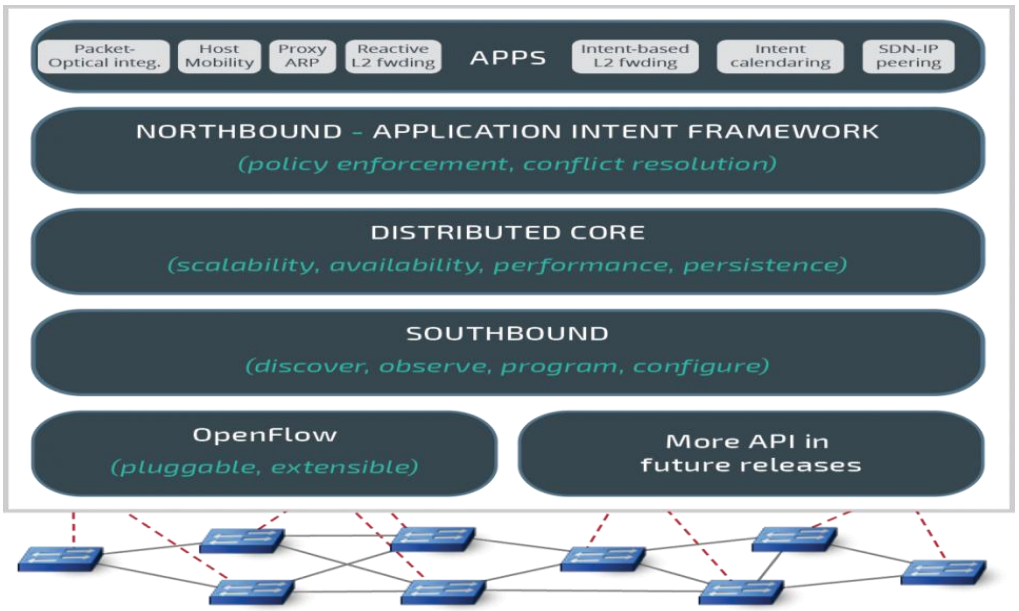

Figure III.1. *Architecture ONOS*

ONOS est composé des éléments suivants [34] :

-Distributed Core : ONOS est déployé en tant que service sur un cluster de serveurs Le même logiciel ONOS s'exécute sur chaque serveur. La symétrie de l'expansion est Considérations de conception importantes pour permettre un basculement rapide en cas de panne Défaillance du serveur ONOS. Les opérateurs de réseau peuvent ajouter des serveurs Augmenter la capacité progressivement et sans interruption selon les besoins plan de contrôle.

Les instances ONOS fonctionnent ensemble pour créer ce que vous voyez Utilisez le reste de votre réseau et de vos applications comme une seule plateforme Application De plus, les périphériques réseau n'ont pas besoin de savoir si un seul périphérique fonctionne. Instance ou plusieurs instances d'ONOS. Cette fonctionnalité rend ONOS évolutif. La capacité d'ONOS évolue de manière transparente C'est le noyau distribué.

- Northbound abstraction/APIs : The Global Network View fournit à l'application une vue du réseau les hôtes, les commutateurs, les liens et tout autre état associé au réseau tel que l'utilisation. Les applications peuvent programmer cette vue de réseau via une API. L'API permet Une application qui présente une vue sous forme de diagramme de réseau. Voilà quelqu'un exemple de ce qu'un diagramme de réseau peut faire :

a) Construire une application Puisque l'application a déjà des vues, calculer le chemin le plus court est facile Diagramme de réseau
b) Maximisez l'utilisation du réseau en surveillant la vue du réseau Modifier le chemin de programmation pour ajuster la charge (ingénierie du trafic)

-Southbound abstraction/APIs : L'abstraction Sud est créée à l'aide d'éléments de réseau, commutateur, hôte ou lien. L'abstraction ONOS de southern représente Chaque élément de réseau est traité comme une forme générique d'objet. L'abstraction permet aux noyaux distribués de maintenir l'état des éléments de réseau sans avoir à le maintenir. Connaît les détails de l'élément représenté par le pilote sous-jacent. Le principal avantage de l'abstraction Southern est qu'elle permet de gérer Différents appareils et différents protocoles sans effet sur le noyau distribué du système et la capacité d'ajouter de nouveaux dispositifs et protocoles au système.

-Software modularity : La création des logiciels est importante. Un logiciel bien conçu est facile à utiliser Améliorer, réparer et entretenir. L'équipe ONOS a travaillé avec beaucoup de soin la modularité permet aux développeurs de travailler plus facilement avec les logiciels. Les logiciels est structuré en composants et comment ils sont structurés les composants sont interconnectés. Comme le montre la figure ci-dessous, la structure principale d'ONOS est constituée de ses couches, qui s'articulent autour de cœurs distribués. Il y a La modularisation des logiciels présente de nombreux avantages :

a) Intégrité et cohérence architecturales
b) Structure de test simplifiée, permettant des tests plus complets
c) Une maintenance plus facile avec moins d'effets secondaires des changements - Extensibilité et personnalisation des composants
d) Éviter les dépendances cycliques

III.3 Topologie de démonstration

La figure III.2 illustre la topologie configurée via le fichier « **topo.py** » contenu dans le dossier du didacticiel « **/onos/tools/tutorials/artemis/topo.py** ». Les haut-parleurs BGP sont des routeurs Quagga et le collecteur de routes est un routeur ExaBGP exécutant un script personnalisé pour reproduire le comportement d'un collecteur de routes RIPE RIS [18].

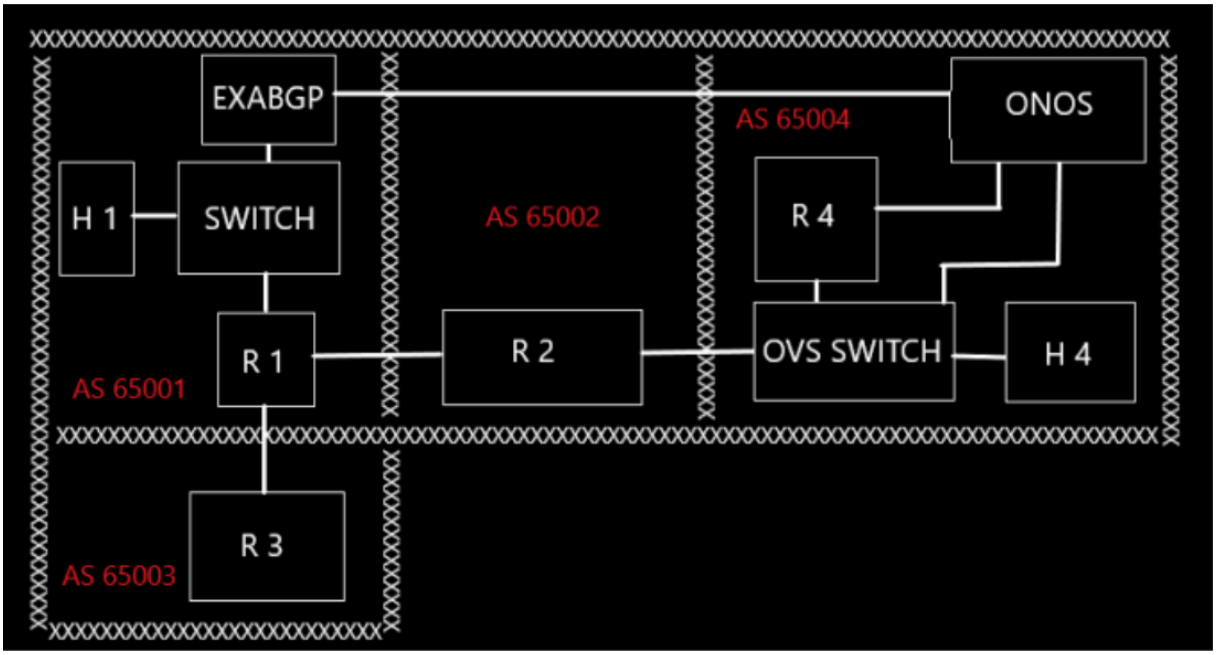

Figure III.2. *Topologie de démonstration conceptuelle*

III.3.1 AS Intermédiaire AS65001

AS intermédiaire se compose d'un haut-parleur BGP (R1), d'un commutateur (L2), d'un hôte (H1) et d'un collecteur de routes ExaBGP (RC).

1. Haut-parleur BGP (R1)

Il annonce via BGP le préfixe 10.0.0.0/8 à ses voisins, en l'occurrence, AS65002 et AS65003. De plus, il a une session iBGP établie avec l'exaBGP RC afin qu'il lui propage des messages de mise à jour BGP, et que exaBGP agisse comme un service de surveillance BGP.

2.Collecteur de routes ExaBGP (RC)

Il est connecté à R1 via iBGP et au contrôleur ONOS sur l'AS protégé. Cette dernière connexion peut être établie via le réseau existant (un tunnel). La seule limitation est que l'interface réseau de l'ONOS qui s'interconnecte avec le RC doit avoir une adresse IP non détournée attribuée afin qu'elle puisse être atteinte par le service de surveillance pendant le détournement.

3. Hôte (H1 / 10.0.0.100)

Cet hôte communique avec l'hôte à l'intérieur de l'AS protégé. Il est utilisé pour nous fournir une visualisation du comportement du plan de données lorsque le détournement BGP se produit.

III.3.2 AS Intermédiaire AS65002

Il se compose d'un haut-parleur BGP (R2) qui annonce via BGP le préfixe 20.0.0.0/8 à ses voisins (R1, R4). Son but est d'ajouter un saut supplémentaire à l'AS-PATH afin que le protégé AS puisse être détourné. Bien que l'attaquant annonce le préfixe exact appartenant à l'AS protégé, le pirate est capable de détourner le trafic provenant de AS65001 vers IP adresses du préfixe détourné.

III.3.3 AS Hijacker AS65003

Il se compose d'un haut-parleur BGP (R3). En annonçant le préfixe de l'AS protégé (40.0.0.0/8) de ce haut-parleur BGP, nous déclenchons un détournement BGP. Tout le trafic généré à partir d'AS65001 et dirigé vers AS65004, sera redirigé vers le réseau d'AS65003.

III.3.4 AS protégé AS65004

Il se compose d'un haut-parleur BGP (R4), d'un commutateur OVS, d'un hôte (H4) et de l'instance ONOS.

1. Haut-parleur (R4)

Il annonce au BGP 40.0.0.0/8. Il est connecté à son voisin (R2) via le commutateur OVS via l'application SDN-IP.

2. OVS

Il communique avec ONOS sur une interface de gestion via 192.168.0.0/24.

3. ONOS

Il est connecté à R4 pour récupérer la table de routage BGP. En outre, il reçoit les messages de mise à jour BGP de l'ExaBGP RC, lorsque des modifications de routage se produisent. Enfin, il est connecté au commutateur OVS afin d'interagir avec le plan de données.

4. H4 / 40.0.0.100

C'est l'hôte qui reçoit le trafic à l'aide de l'application de routage réactif de l'hôte dans AS65001 (H1).

III.4 Installation et configuration

Dans cette section, nous détaillons les étapes nécessaires que nous avons suivies pour installer les logiciels requis pour la mise en œuvre de l'ARTEMIS en tant qu'application multimodules fonctionnant sur ONOS [35].

Dans un premier temps, nous présentons une configuration propre d'une machine virtuelle Ubuntu 20.04, avec au moins 40 Go d'espace HD, deux processeurs, 8 Go de RAM et une seule interface NATted.

III.4.1 ExaBGP Python

« ExaBGP » est une nouvelle application conçue pour fournir aux programmeurs et aux administrateurs système un moyen simple d'interagir avec les réseaux BGP. Le programme est conçu pour permettre l'injection des routes arbitraires dans un réseau.

« ExaBGP » est un outil polyvalent qui convient le mieux pour :

- Fournir des solutions de basculement entre les Data Center migrant / 32 services IP.

- Atténuer les attaques de réseau

- Déployer de manière centralisée des filtres au niveau du réseau (trou noir et / ou flux spécifique)

- Collecter les informations sur le réseau à l'aide de BGP-LS ou BGP avec Add-Path

On lance un terminal et on installe « ExaBGP Python » :

```
$ cd ~
$ sudo apt-get install git
$ git clone -b 3.4 https://github.com/Exa-Networks/exabgp
$ echo 'export PATH=$PATH:~/exabgp/sbin' >> ~/.bashrc
$ source ~/.bashrc
```

Figure III.3. *Installation de « ExaBGP »*

III.4.2 Installation logiciel « Quagga »

« Quagga » est une suite de logiciels de routage implémentant les protocoles OSPF, RIP, BGP et IS-IS pour les plates-formes de type Unix, en particulier, FreeBSD et GNU/Linux.

```
$ cd ~
$ sudo apt-get install quagga -y
```

Figure III.4. *Installation « Quagga »*

III.4.3 Téléchargement et installation « 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 extrêmement flexible et un réseau défini par logiciel [36] et [37].

```
$ git clone -b 2.3.0 --single-branch git://github.com/mininet/mininet
$ PYTHON=python3 mininet/util/install.sh -nfv
```

Figure III.5. *Installation « Mininet »*

III.4.4 ONOS

Afin d'installer et exécuter ONOS, de nombreuses packages sont recommandés, à savoir :

- JAVA

- Bazel

- ONOS

- Curl

Pour les installer, il suffit d'utiliser les commandes suivantes :

1- Installation de Java11 comme suggéré par le wiki d'ONOS.

```
  $ cd ~
  $ sudo apt-get install software-properties-common -y && \
> sudo add-apt-repository ppa:webupd8team/java -y && \
> sudo apt-get update && \
> echo "oracle-java11-installer shared/accepted-oracle-licebse-v1-1. select true" | sudo debconf-set-selections && \
> sudo apt-get install oracle-java11-installer oracle-java11-set-default -y
```

Figure III.6. *Installation Java*

Utiliser JDK 11 comme JDK principale :

```
$ sudo apt install openjdk-11-jdk
$ export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
$ wget https://inmon.com/products/sFlow-RT/sflow-rt.tar.gz
$ tar -xvzf sflow-rt.tar.gz
$ ./sflow-rt/start.sh
```

Figure III.7. *Basculer vers JDK 11*

« Update-java » est contenu à l'intérieur « java-commonet » est très utile pour basculer JDK11 et utiliser comme JDK principaux.

2- Installation de Bazel et d'autres dépendances

```
$ sudo apt-get install ssh git curl zip décompresser python python3 bzip2
$ sudo apt-get install pkg-config g ++ zlib1g-dev
```

Figure III.8. *Installation package dépendant par Bazel*

Bazel est utilisé pour compiler, tester et générer la plupart des applications. Il est conçu pour construire et exécuter ONOS.

Pour le téléchargement et l'installation de Bazel, nous avons utilisé le site « https://github.com/bazelbuild/bazel/releases » pour télécharger le fichier bazel-3.7.2-installerlinux-x86_64.sh.

Nous avons exécuté les commandes suivantes :

```
$ cd
$ mkdir -p Téléchargements
$ wget https://github.com/bazelbuild/bazel/releases/download/3.7.2/bazel-3.7.2-installer-linux-x86_64.sh
$ cd ~/Téléchargements
$ chmod + x bazel-3.7.2-installer-linux-x86_64.sh
$ ./bazel-3.7.2-installer-linux-x86_64.sh -- user
```

Figure III.9. *Installation Bazel*

Il faut mettre à jour les variables d'environnements suivantes :

```
$ echo 'export PATH = "$PATH:$HOME/bin" ' >> ~/.bashrc
$ echo 'source $HOME/.bazel/bin/bazel-complete.sh' >> ~/.bashrc
$ source ~/.bashrc
```

Figure III.10. *Configurations des variables d'environnement Bazel*

3- Installation ONOS

- Install bazelisk [38]
- Pour ONOS, nous avons téléchargé ONOS depuis « GitHub » et nous avons configuré le profil « bash » correspondant, comme illustré par la figure :

```
$ wget https://github.com/bazelbuild/bazelisk/releases/download/v1.7.5/bazelisk-linux-amd64
$ chmod +x bazelisk-linux-amd64
$ sudo mv bazelisk-linux-amd64/usr/local/bin/bazel
```

Figure III.11. *install bazelisk*

```
$ git clone -b 2.5.1 --single-branch https://github.com/opennetworkinglab/onos
$ export ONOS_ROOT=~/onos
$ source $ONOS_ROOT/tools/dev/bash_profile
```

Figure III.12. *Téléchargement ONOS*

- Build ONOS from source

```
$ cd $ONOS_ROOT
$ bazel build onos
```

Figure III.13. *Lancement de Build*

Build ONOS with Bazel

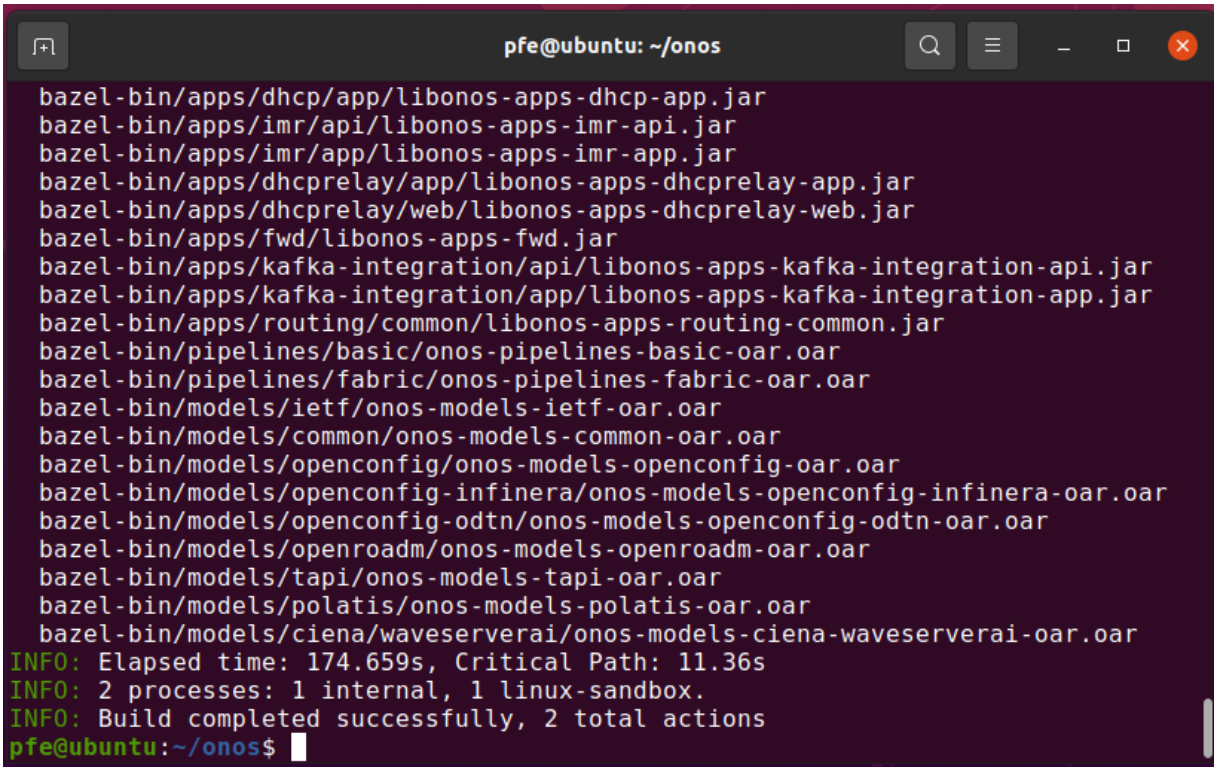

Figure III.14. *Bazel Build Onos*

III.4.5 Installation Pip3

Nous avons installé « pip3 », packages Python et avons défini la configuration utilisée par « ExaBGP » :

```
$ sudo apt-get install python3-pip -y
$ sudo -H pip3 install -r ~/onos/tools/tutorials/artemis/requirements.txt
$ cd ~/onos/tools/tutorials/artemis/
$ sed -i 's?/absolute/path/to/onos/tools/tutorials/artemis?'`pwd`'?' configs/exabgp.conf
```

Figure III.15. *Installation Pip3*

III.5 Tests de l'application ARTEMIS

III.5.1 Exécuter ONOS et charger la topologie « Mininet »

Pour exécuter ONOS, nous avons exécuté la commande ci-dessous dans le terminal. Il est à noter que la première exécution prend un certain temps.

```
$ sudo apt-get install curl
$ cd onos
$ bazel run onos-local --clean
```

Figure III.16. *Run ONOS*

Après le démarrage de l'instance ONOS, nous ouvrons un nouveau terminal et chargeons la topologie « mininet » via le fichier « topo.py » dans le dossier du didacticiel :

« /onos/tools/tutorials/artemis/topo.py »

```
$ cd onos/tools/tutorials/artemis
$ sudo ./artemis-topo.py
```

Figure III.17. *Lancement de topologie*

Nous avons deux Terminaux ouverts, en l'occurrence, instance ONOS et « Mininet », comme indiqué sur la figure III.15 :

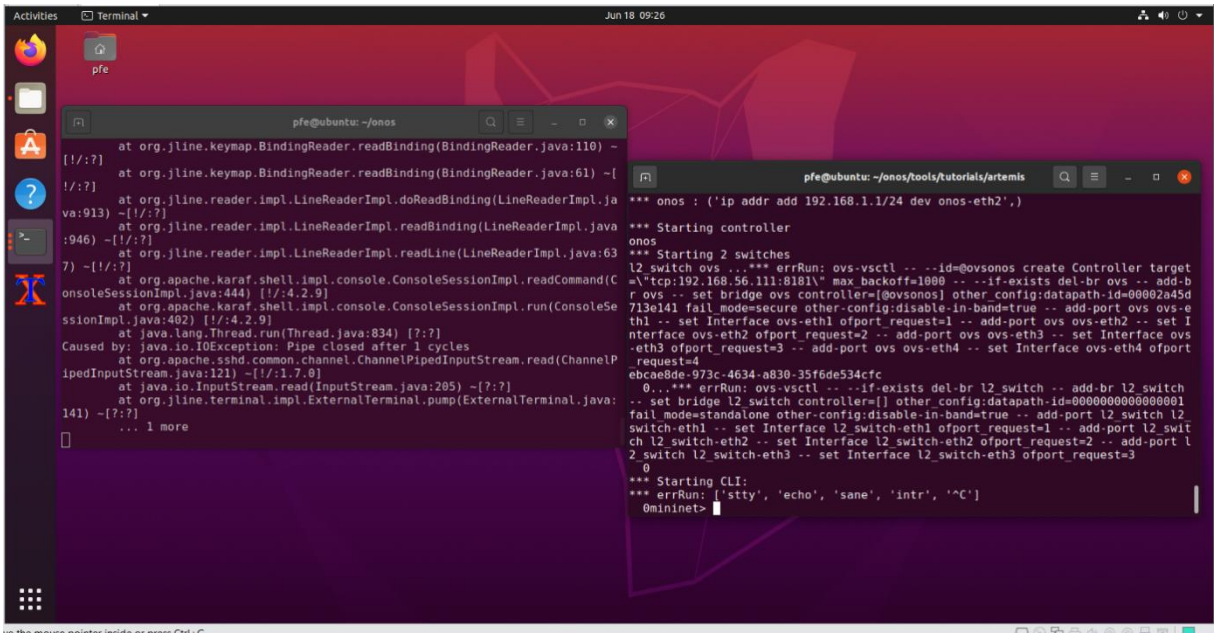

Figure III.18. *Instance ONOS et « Mininet »*

III.5.2 Lancer ONOS et activer ARTEMIS

Nous ouvrons un nouveau terminal et chargeons la configuration réseau avec « onos-netcfg ». Depuis ce terminal, on se connecte à la « CLI ONOS » :

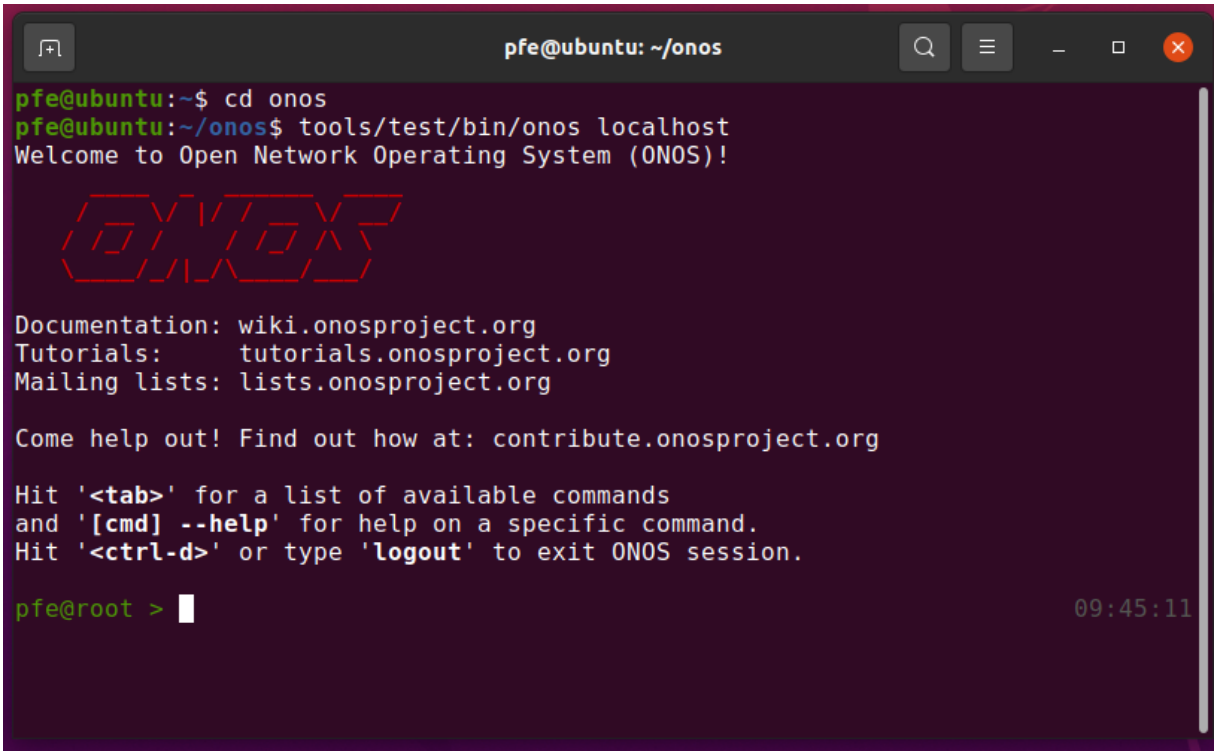

Figure III.19. *Lancement CLI ONOS*

Nous avons activé ARTEMIS à l'aide de la « CLI ONOS ». Le routage réactif doit d'abord être activé :

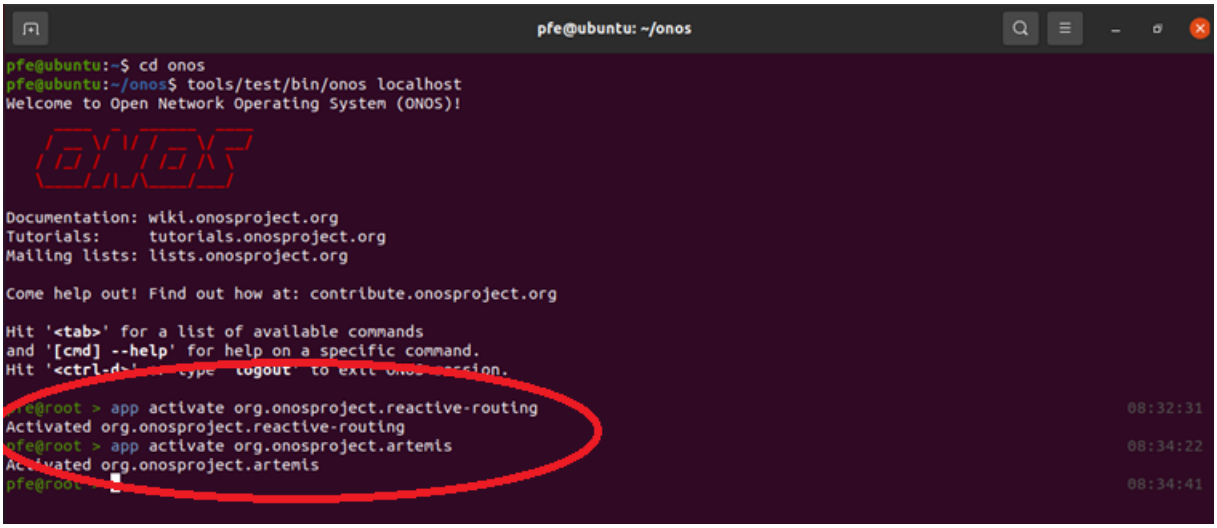

Figure III.20. *Activation ARTEMIS*

Il est à noter que les protocoles de routage réactifs, dits aussi protocoles de routage à la demande, représentent les protocoles les plus récents proposés dans le but d'assurer le service du routage dans les réseaux sans fils.

Les protocoles de routage appartenant à cette catégorie, créent et maintiennent les routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte globale de routes est lancée, et cela dans le but d'obtenir une information.

III.5.3 Etablir une connexion entre hôtes/détourner le préfixe de l'AS protégé

1- Via la « CLI mininet », on fait un ping entre les hôtes H1 et H4 pour établir une connexion entre AS65001 et AS65004

Figure III.21. *Ping entre H1 et H4*

Il est à noter que la commande « pingall » est utilisée pour rendre les hôtes visibles et la commande « xterm R3 » permet d'ouvrir un nouveau terminal sur R3.

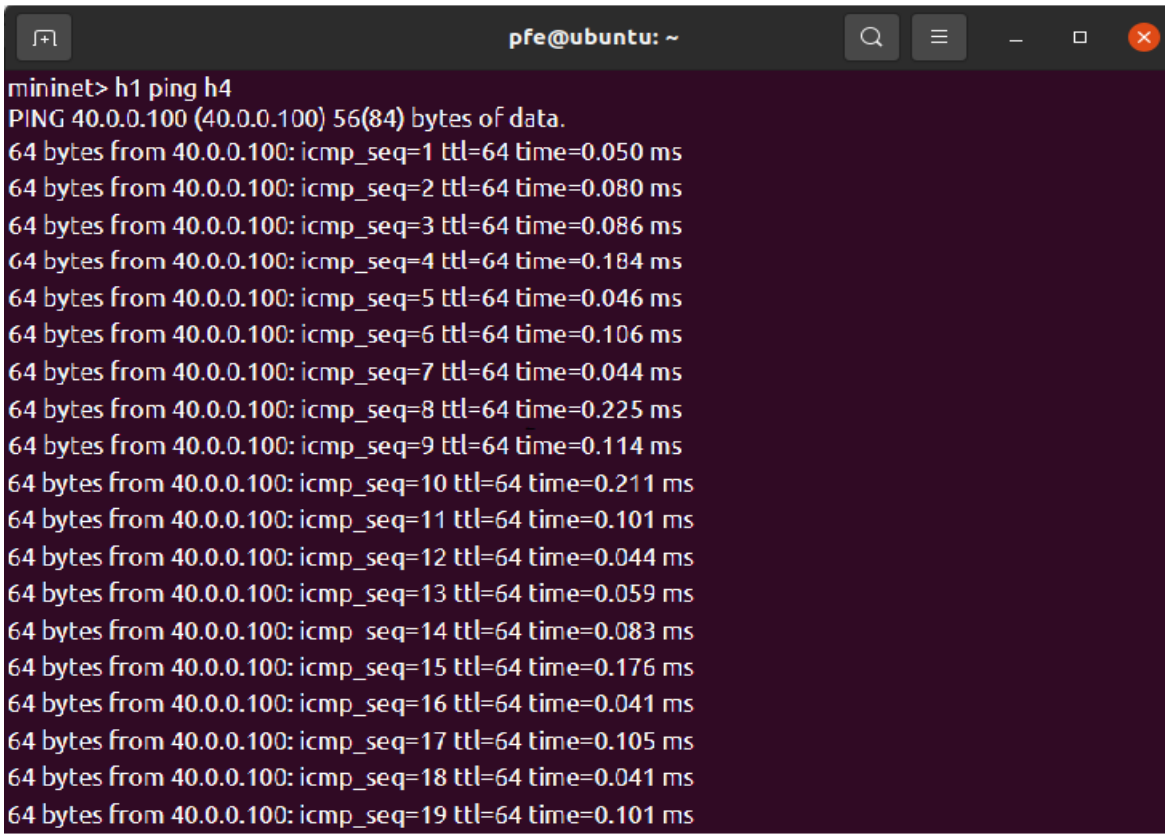

Figure III.22. *Résultat de Ping*

2- Pour détourner le préfixe de l'AS protégé, nous avons annoncé le préfixe 40.0.0.0/8, préfixe de l'AS protégé, sur le terminal de R3.

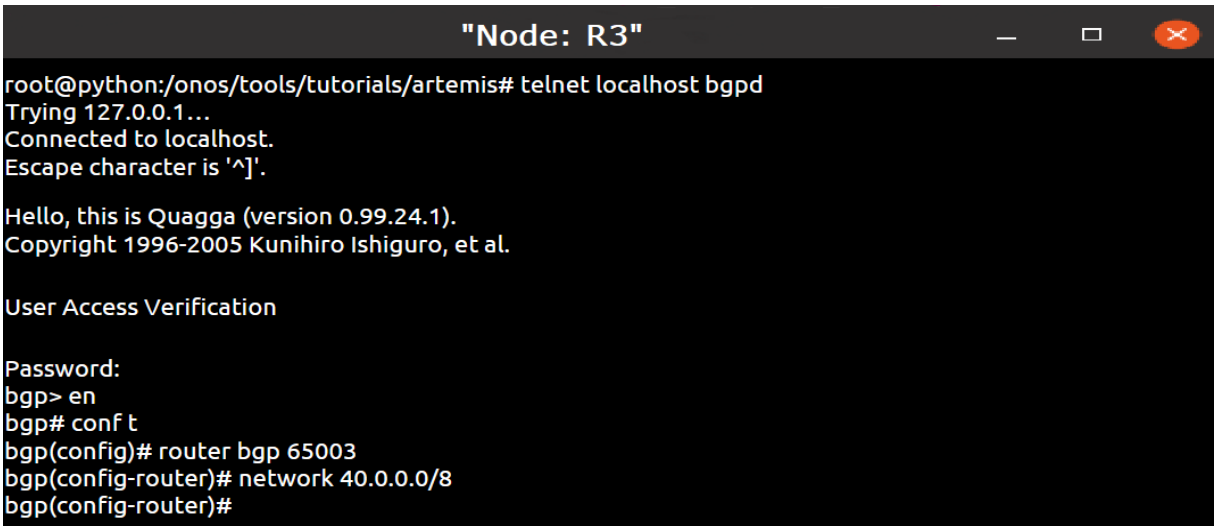

Figure III.23. *Détournement du préfixe de l'AS protégé*

Le pirate de l'air (AS65003) a attiré tout le trafic de AS65001, destiné à 40.0.0.0/8, loin de AS65004. En même temps, le haut-parleur « ExaBGP » dans AS65001 a envoyé la mise

à jour « BGP » du détournement, parmi d'autres mises à jour vues par AS65004, à l'instance ONOS, exécutant ARTEMIS.

Le détournement a été détecté. On a vu que le ping s'est arrêté pendant quelques secondes, avec une perte de données de seulement 4 paquets, puis il a redémarré.

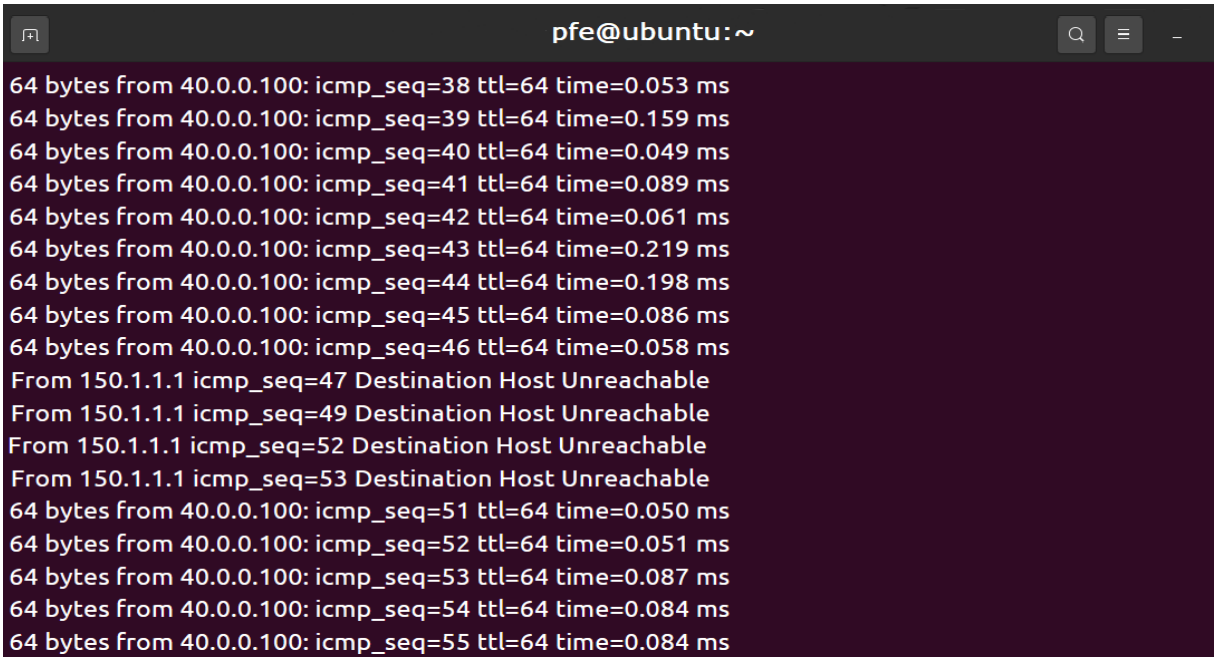

Figure III.24. *Détection de l'attaque*

Nous avons vu que l'attaque est réellement détectée et que le mécanisme de désagrégation a réussi à atténuer l'attaque. Depuis R4, nous avons observé qu'ARTEMIS a automatiquement configuré R4 pour annoncer les préfixes les plus spécifiques 40.0.0.0/9 et 40.128.0.0/9, comme la montre la figure III.25 :

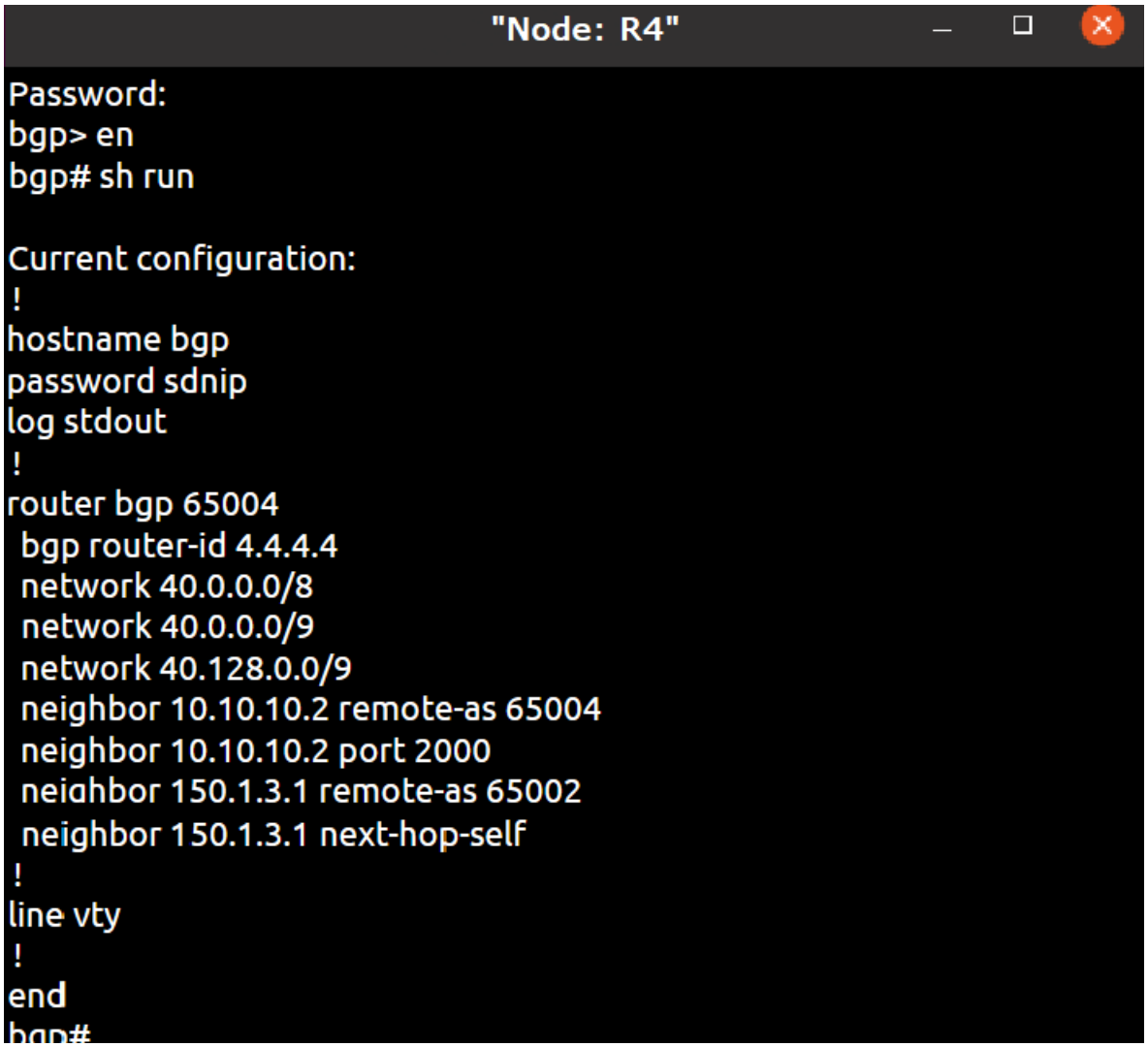

Figure III.25. *Préfixes spécifiques*

III.6 Conclusion

Au cours de ce chapitre, nous avons abordé la mise en pratique de notre projet. Nous avons utilisé la technologie SDN, en particulier le contrôleur ONOS, pour mettre en œuvre notre application baptisée "ARTEMIS". Cette application vise à protéger contre les attaques de détournement de préfixe BGP.

Cette étape nous a permis d'approfondir nos connaissances sur les réseaux SDN, le routage Internet et plus spécifiquement le protocole BGP, ainsi que leur impact sur la sélection des chemins.

Conclusion Générale

Source

« Car il est parfois facile, une fois qu'on
Acquiert une certaine connaissance des
Questions d'en imaginer, ensuite, la
Démonstration que si l'on cherchait sans
Aucune notion préalable »

-Archimède de Syracuse-

Conclusion générale

Le SDN « Software Defined Networking » désigne une approche novatrice de la programmabilité réseau, offrant la possibilité d'initialiser, de contrôler, de modifier et de gérer dynamiquement le comportement du réseau grâce à des interfaces ouvertes. Le réseau SDN propose diverses applications pour les administrateurs réseau, permettant ainsi la surveillance et le contrôle automatique des réseaux.

Le réseau SDN permet de résoudre les problèmes de l'augmentation, la complexité et le manque de contrôle dans le réseau Internet. De plus, il fournit des mécanismes de sécurité contre plusieurs types de risques d'attaque existant sur Internet. Il permet de protéger et contrôler le trafic dans le réseau pour éviter tous les incidents qui peuvent conduire à des problèmes de routage et des pertes économiques.

Les contrôleurs SDN permettent une gestion centralisée des réseaux, une utilisation optimisée grâce au calcul de chemin centralisé, une surveillance en temps réel des données et une mise à l'échelle de la capacité du réseau.

Dans le contexte des interconnexions Internet, les opérateurs et administrateurs réseau doivent garantir des liaisons fiables avec les autres opérateurs qui utilisent généralement le protocole BGP. Cependant, ces interconnexions sont souvent exposées aux risques liés au manque de contrôle et de surveillance sur le réseau Internet accessible à tous. Par conséquent, il est possible qu'un opérateur annonce des chemins ou préfixes illégitimes appartenant à d'autres opérateurs, ce que l'on appelle le détournement de préfixe. Les contrôleurs SDN offrent des outils de détection et de protection contre ces incidents.

C'est précisément en raison de tous ces avantages potentiels que cette technologie a suscité notre intérêt et est devenue l'objet de notre travail. Nous nous sommes focalisés sur l'une des applications proposées par le contrôleur ONOS, nommée "ARTEMIS". Cette application permet la détection en temps réel et l'atténuation automatique des incidents de détournement de préfixe affectant les préfixes sous contrôle administratif. Elle utilise l'autosurveillance des données BGP, telles que les mises à jour BGP exportées par les collecteurs de routes.

Les résultats obtenus sont concluants, répondant ainsi aux questions posées et aux contraintes identifiées dans la problématique. Nous avons constaté l'impact positif de

l'application "ARTEMIS" sur l'évolution et la gestion efficace des réseaux d'opérateurs. Nous avons convaincu des bénéfices apportés par cette nouvelle solution.

Ce projet nous a permis principalement d'approfondir nos connaissances en matière de l'évolution des réseaux d'opérateurs. Il nous a également offert l'opportunité d'appliquer les principes et les méthodes théoriques acquis au cours de notre parcours universitaire.

Cependant, le développement d'un tel projet n'est jamais totalement achevé et certaines idées n'ont pas pu être réalisées en raison des contraintes de temps. En guise de perspectives, il serait envisageable d'étendre la prise en charge à plusieurs routeurs de fournisseurs, au-delà de Quagga, en utilisant des protocoles tels que NETCONF et YANG. De plus, l'utilisation de GNS3 comme plateforme d'émulation pourrait être explorée.

En conclusion, nous espérons que ce travail servira de base pour les futurs étudiants afin qu'ils puissent apporter des améliorations ultérieures, car l'évolution des réseaux d'opérateurs ouvre de nombreuses opportunités à explorer.

Références

Bibliographique

Recherche

« Un chercheur qui cherche, on en trouve

Un chercheur qui trouve, on en cherche »

[1] « What is the control plane ? | Control plane vs. Data plane » . Site web : https://www.cloudflare.com (consulté le 15/02/2023)

[2] P. Rollin, « **Réseaux Locaux : Normes et Protocoles** », Edition Hermès, 1993.

[3] « Routage Dynamique : Protocoles de routage dynamique ». Site web : https://www.formip.com (consulté le 15/02/2023)

[4] Han Ligang, « **DATA COMMUNICATIONS AND NETWORK TECHNOLOGIES** », Huawei Technologies Co., Ltd, December 2021, ISBN 978-981-19-3029-4, p195.

[5] Han Ligang, « **DATA COMMUNICATIONS AND NETWORK TECHNOLOGIES** », Huawei Technologies Co., Ltd, December 2021, ISBN 978-981-19-3029-4, p230.

[6] Collectif cisco, **« Cisco networking academy program: CCNA 1 and 2 companion guide** », 3rd edition, 2005, ISBN 1-58713-150-1, p697.

[7] « **Le SDN pour les nuls ».** Site web : https://gblogs.cisco.com (consulté le 19/02/2023).

[8] « Simple Network Management Protocol (SNMP) ». Site web : http://searchnetworking.teletarget.com (consulté le 23/02/2023)

[9] « Le but de SDN ». Site web : https://www.lemagit.fr (consulté le 27/02/2023]

[10] Jérôme Durand, **« Le SDN pour les nuls** », Montpellier, JRES 2015.

[11] Traore Issa, Kouassi Brou Médard, et Atta Ferdinand, « **Etude du nomadisme dans un Cloud éducatif administré par la technologie SDN/OpenFlow** », Institut de recherches mathématique Université Félix Houphouët-Boigny, conférence WACREN 2016.

[12] « Interface SouthBound ». Site web : https://whatis.techtarget.com (consulté le 02/03/2023)

[13] H. BOUIDA, '**Etude et mise en oeuvre d'une solution SDN** : Application de Gestion de VLANs', (2017, Janvier,18) Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat – Maroc.

[14] Thomas Paradis, « **Software-Defined Networking** », mémoire de Master, School of Information and Communication Technology KTH Royal Institute of Technology Stockholm, Sweden, le 20 Janvier 2014.

[15] « Junos® OS OpenFlow User Guide ». Site web : https://www.juniper.net (consulté le 05/03/2023)

[16] « OpenFlow Working Mechanism ». Site web : https://forum.huawei.com (consulté le 05/03/2023)

[17] « What Is OpenFlow? How Does It Relate to SDN? ». Site web : https://forum.huawei.com (consulté le 06/03/2023)

[18] « Guide de configuration de la programmabilité ». Site web : https://www.cisco.com (consulté le 08/03/2023)

[19] Steven J. Vaughan-Nichols, « **OpenFlow: The Next Generation of the Network?** », Published by the IEEE Computer Society, AUGUST 2011, p14.

[20] Y Rekhter T.LI S. Hares, « **RFC 4271: A Border Getway Protocol 4** », 2006.

[21] Clément SAAD, « **Instabilités du protocole BGP** », Memoire de D.E.A, Université Montpellier, 2015.

[22] « **About PEERING - The BGP Testbed** ». Site : https://peering.usc.edu/

[consulté le 25/03/2023]

[23] F. Contat, S. Nataf & G. Valadon. « **Influence des bonnes pratiques sur les incidents BGP** ». Agence Nationale de la Sécurité des Systèmes d'Information. Juin 2012

[24] « What is BGP Hijacking ? ». Site web : https://www.purevpn.com (consulté le 28/03/2023)

[25] F. Contat, S. Nataf & G. Valadon. « **Influence des bonnes pratiques sur les incidents BGP** ». Agence Nationale de la Sécurité des Systèmes d'Information. Juin 2012.

[26] Pierre-Antoine VERVIER, « **Attaques par détournement BGP malveillant : détection, analyse et contre-mesures** », thèse pour obtenir le grade de docteur délivré par TELECOM ParisTech, 19 décembre 2014.

[27] « Qu'est-ce que le détournement de BGP ? ». https://www.cloudflare.com (consulté le 15/04/2023)

[28] « Net boffins brew poison for BGP hijacks ». Site web : https://www.theregister.com (consulté le 17/04/2023)

[29] « ARTEMIS : Neutralising BGP Hijacking Within a Minute ». Site web : https://labs.ripe.net (consulté le 18/04/2023)

[30] « About PEERING - The BGP Testbed ». Site web : https://peering.usc.edu (consulté le 22/04/2023)

[31] P. Sermpezis, G. Chaviaras, P. Gigis, and X. Dimitropoulos « **Monitor, Detect, mitigate: Combating BGP Prefix Hijacking in Real-Time with ARTEMIS** ». FORTH, Greece. Septembre 2016.

[32] « ARTEMIS: an Automated System against BGP Prefix Hijacking ». Sie web : https://wiki.onosproject.org (consulté le 02/05/2023)

[33] C. Chaviaras, P. Gigis, P. Sermpezis & X. Dimitropoulos. « **ARTEMIS: Real-Time Detection and Automatic Mitigation for BGP Prefix Hijacking** ». Forth / University of Crete, Greece. 2016

[34] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor & P. Radoslavov. « **ONOS : Towards an open, distributed SDN OS** ». HotSDN '14 : Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, pp. 1-6. August 2014.

[35] T. Vachuska, J. Halterman, A. Campanella, B. O'Connor, D. Bainbridge, R. Milkey & C. Cascone. « **Network Operation System Designed for High Availability, Perormance, Scale-out** ». ONF / Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science, INSPIRE group. 2017.

[36] D. Mavrommatis, L. Manassakis & V. Kotronis. « **ARTEMIS : an Automated System against BGP Prefix Hijacking** ». ONF / Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science, INSPIRE group. April 2018.

[37] I.Z. Bholebawa, R.K. Jha & U.D. Dalal. « **Performance Analysis of Proposed OpenFlow-Based Network Architecture Using Mininet** ». Wireless Personal Communications, Vol. 86, pp. 943-958. July 2017

[38] P. Danielis, V. Altmann, J. Skodzik, E.B. Schweissguth, F. Golatowski & D. Timmermann. « **Émulation de réseaux d'automatisation pris en charge par SDN** ». 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). Luxembourg. Septembre 2015.

Annexes

Résultats

« Il ne s'agit pas d'atteindre un résultat vrai ou faux, probable ou improbable, mais seulement profitable ou non profitable »

-Williams & Lance-

Annexe I : Etablissement d'une connexion commutateur-contrôleur

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

A.1. *Cas n°1*

Tout d'abord, il faut renseigner l'adresse IP du contrôleur au niveau du commutateur. Il peut y avoir redondance lors du démarrage du commutateur OpenFlow. Ce dernier envoie un paquet « *OFPT_HELLO* » avec le numéro de version d'OpenFlow supportée.

Le contrôleur vérifie la version d'OpenFlow supportée par le commutateur et lui répond par un message « *OFPT_HELLO* » en indiquant la version d'OpenFlow avec laquelle ils communiqueront. La connexion est ainsi établie.

La figure A.1 présente les principales étapes de la connexion entre le contrôleur et le commutateur OpenFlow.

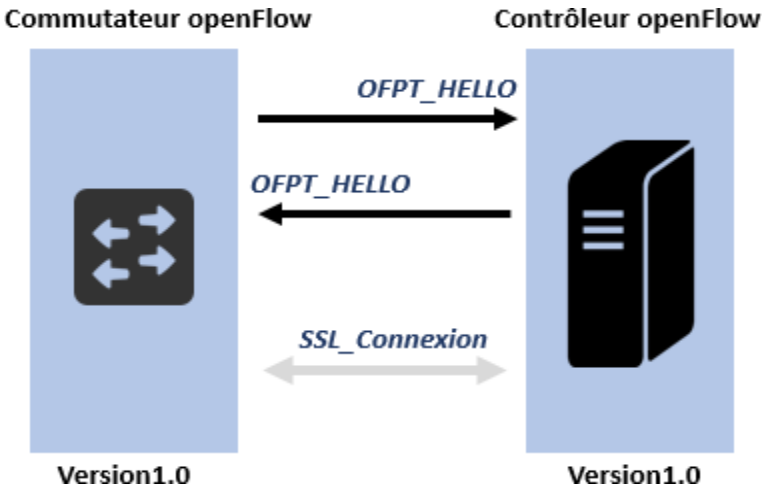

Figure A.1. Connexion au contrôleur OpenFlow

A.2. *Cas n°2*

Comme dans le cas précédent, le commutateur OpenFlow envoie un paquet « *OFPT_HELLO* » avec la version du protocole utilisé, le contrôleur s'aperçoit qu'il ne supporte pas la version OpenFlow du commutateur. Il lui retourne donc un paquet « *OFPT_ERROR* » en indiquant que c'est un problème de compatibilité, comme illustré dans la figure A.2 :

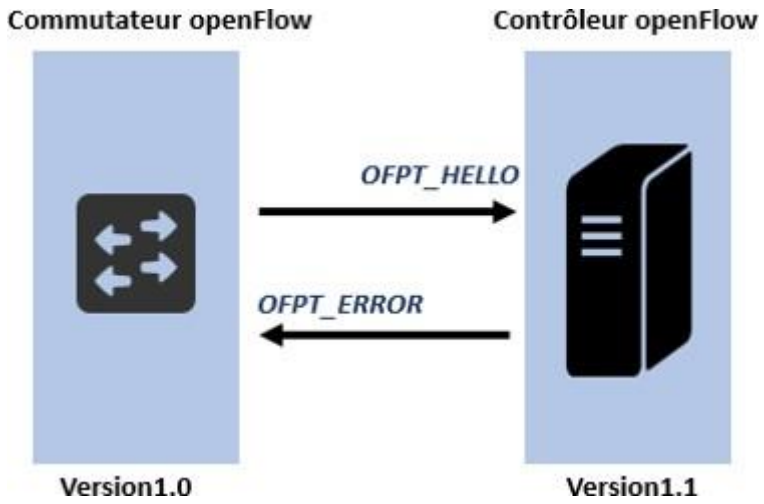

Figure A.2. Échec de la connexion au contrôleur OpenFlow

A.3. *Cas n°3*

Comme dans les cas précédents, le commutateur envoie un paquet « *OFPT_HELLO* » au contrôleur. Si celui-ci ne répond pas, il tente alors de joindre les éventuels autres contrôleurs qui lui ont été paramétrés. S'il ne parvient pas à les joindre, il se met alors en mode urgence « *EMERGENCY-MODE* ». Le commutateur utilise sa table de flux par défaut, si toutefois un paquet ne correspond à aucun enregistrement dans la table alors il le supprime, comme le montre la figure A.3 :

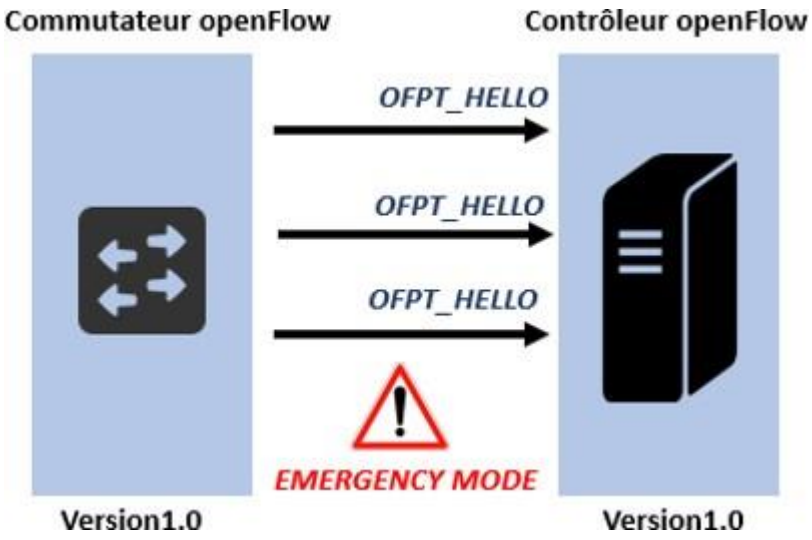

Figure A.3. Mode d'urgence.

Annexe II : Création de la topologie ARTEMIS « artemis-topo.py »

Ci-dessous le ficher correspondant à la création de

notre topologie : #! /usr/bin/python

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.log import setLogLevel, info, debug
from mininet.node import Host, RemoteController, OVSSwitch
import os


QUAGGA_DIR = '/usr/lib/quagga'
# Must exist and be owned by quagga user (quagga:quagga by default on
Ubuntu)
QUAGGA_RUN_DIR = '/var/run/quagga'
EXABGP_RUN_EXE = '~/exabgp/sbin/exabgp'
CONFIG_DIR = 'configs/'


onos = RemoteController('onos', ip='192.168.0.1', port=6633)


class Onos(Host):


    def __init__(self, name, intfDict, *args, **kwargs):
        Host.__init__(self, name, *args, **kwargs)


        self.intfDict = intfDict


    def config(self, **kwargs):
        Host.config(self, **kwargs)


        for intf, attrs in self.intfDict.items():
            self.cmd('ip addr flush dev %s' % intf)
            if 'mac' in attrs:
                self.cmd('ip link set %s down' % intf)
                self.cmd('ip link set %s address %s' % (intf,
attrs['mac']))
                self.cmd('ip link set %s up ' % intf)
            for addr in attrs['ipAddrs']:
                self.cmd('ip addr add %s dev %s' % (addr, intf))
```

```
class QuaggaRouter(Host):



    def __init__(self, name, quaggaConfFile, zebraConfFile, intfDict,
*args, **kwargs):
        Host.__init__(self, name, *args, **kwargs)









        self.quaggaConfFile = quaggaConfFile
        self.zebraConfFile = zebraConfFile
        self.intfDict = intfDict


    def config(self, **kwargs):
        Host.config(self, **kwargs)
        self.cmd('sysctl net.ipv4.ip_forward=1')


        for intf, attrs in self.intfDict.items():
            self.cmd('ip addr flush dev %s' % intf)
            if 'mac' in attrs:
                self.cmd('ip link set %s down' % intf)
                self.cmd('ip link set %s address %s' % (intf, attrs['mac']))
                self.cmd('ip link set %s up ' % intf)
            for addr in attrs['ipAddrs']:
                self.cmd('ip addr add %s dev %s' % (addr, intf))


        self.cmd('/usr/lib/quagga/zebra -d -f %s -z %s/zebra%s.api -i
%s/zebra%s.pid' %
                 (self.zebraConfFile, QUAGGA_RUN_DIR, self.name, QUAGGA_RUN_DIR,
self.name))
        self.cmd('/usr/lib/quagga/bgpd -d -f %s -z %s/zebra%s.api -i
%s/bgpd%s.pid' %
                 (self.quaggaConfFile, QUAGGA_RUN_DIR, self.name, QUAGGA_RUN_DIR,
self.name))


    def terminate(self):
```

```
        self.cmd("ps ax | egrep 'bgpd%s.pid|zebra%s.pid' | awk '{print $1}' |
xargs kill" % (
            self.name, self.name))


        Host.terminate(self)

class ExaBGPRouter(Host):


    def __init__(self, name, exaBGPconf, intfDict, *args, **kwargs):
        Host.__init__(self, name, *args, **kwargs)


        self.exaBGPconf = exaBGPconf
        self.intfDict = intfDict


    def config(self, **kwargs):
        Host.config(self, **kwargs)
        self.cmd('sysctl net.ipv4.ip_forward=1')




        for intf, attrs in self.intfDict.items():
            self.cmd('ip addr flush dev %s' % intf)
            if 'mac' in attrs:
                self.cmd('ip link set %s down' % intf)
                self.cmd('ip link set %s address %s' % (intf, attrs['mac']))
                self.cmd('ip link set %s up ' % intf)
            for addr in attrs['ipAddrs']:
                self.cmd('ip addr add %s dev %s' % (addr, intf))


        self.cmd('%s %s > /dev/null 2> exabgp.log &' % (EXABGP_RUN_EXE,
self.exaBGPconf))


    def terminate(self):
        self.cmd(
            "ps ax | egrep 'lib/exabgp/application/bgp.py' | awk '{print $1}' |
xargs kill")
        self.cmd(
            "ps ax | egrep 'server.py' | awk '{print $1}' | xargs kill")
        Host.terminate(self)
```

```
class ONOSSwitch(OVSSwitch):


    def start(self, controllers):
        return OVSSwitch.start(self, [onos])


         def start(self, controllers):
             return OVSSwitch.start(self, [])




     class ArtemisTopo(Topo):
         "Artemis tutorial topology"


         def build(self):
             zebraConf = '%szebra.conf' % CONFIG_DIR


             quaggaConf = '%sR1-quagga.conf' % CONFIG_DIR
             name = 'R1'
             eth0 = {
                 'ipAddrs': ['150.1.1.2/30']
             }
             eth1 = {
                 'ipAddrs': ['10.0.0.1/8']
             }
             eth2 = {
                 'ipAddrs': ['150.1.2.1/30']
             }
             intfs = {
                 '%s-eth0' % name: eth0,
                 '%s-eth1' % name: eth1,
                 '%s-eth2' % name: eth2
             }
              r1 = self.addHost(name, cls=QuaggaRouter, quaggaConfFile=quaggaConf,
                                zebraConfFile=zebraConf, intfDict=intfs)


             quaggaConf = '%sR2-quagga.conf' % CONFIG_DIR
             name = 'R2'
             eth0 = {
                 'ipAddrs': ['150.1.3.1/30']
             }
             eth1 = {
                 'ipAddrs': ['150.1.2.2/30']
             }
```

```
            intfs = {
                '%s-eth0' % name: eth0,
                '%s-eth1' % name: eth1
            }
             r2 = self.addHost(name, cls=QuaggaRouter, quaggaConfFile=quaggaConf,
                               zebraConfFile=zebraConf, intfDict=intfs)


class L2Switch(OVSSwitch):


        eth1 = {
            'ipAddrs': ['192.168.1.2/24']
        }
        intfs = {
            '%s-eth0' % name: eth0,
            '%s-eth1' % name: eth1
        }
        exabgp = self.addHost(name, cls=ExaBGPRouter,
                              exaBGPconf='%sexabgp.conf' % CONFIG_DIR,
                              intfDict=intfs)


        self.addLink(r1, r3, port1=0, port2=1)
        self.addLink(r1, l2_switch, port1=1, port2=2)
        self.addLink(r1, r2, port1=2, port2=1)


        self.addLink(ovs, r2, port1=2, port2=0)
        self.addLink(ovs, h4, port1=3, port2=0)
        self.addLink(ovs, r4, port1=4, port2=0)


        self.addLink(l2_switch, h1, port1=1, port2=0)
        self.addLink(l2_switch, exabgp, port1=3, port2=0)


        name = 'onos'
        eth0 = {
            'ipAddrs': ['192.168.0.1/24']
        }
        eth1 = {
            'ipAddrs': ['10.10.10.2/24']
        }
        eth2 = {
            'ipAddrs': ['192.168.1.1/24']
        }
        intfs = {
            '%s-eth0' % name: eth0,
            '%s-eth1' % name: eth1,
            '%s-eth2' % name: eth2
```

```
            }
            onos = self.addHost(name, inNamespace=False, cls=Onos, intfDict=intfs)


            self.addLink(onos, ovs, port1=0, port2=1)
            self.addLink(onos, r4, port1=1, port2=1)
            self.addLink(onos, exabgp, port1=2, port2=1)


topos = {'artemis': ArtemisTopo}
if __name__== '__main__':
    setLogLevel('debug')
    topo = ArtemisTopo()


    net = Mininet(topo=topo, build=False)
    net.addController(onos)
    net.build()
    net.start()


    CLI(net)


    net.stop()


    info("done\n")
```

ملخص

تتكون الأنترنت من ألاف الأنظمة الذاتية AS حيث توجه حركة المرور باستخدام بروتوكول ال BGP. نظرا للتوزيع الطبيعي ونقص التفويض في BGP، يمكن ل AS الإعلان عن مسارات او بادئات غير مشروعة تنتمي الى AS أخرى، أي تجاوز البادئات الخاصة بها .

تقدم وحدة التحكم ONOS تطبيق " ARTEMIS" لمسؤولي الشبكات الذي يمكن من الكشف في الوقت الحقيقي والتخفيف التلقائي من حوادث الاختراق في المسارات، باستخدام المراقبة الذاتية على مستوى AS.

الكلمات المفتاحية: بادئات، ARTEMIS, ONOS, BGP, AS

Résumé

L'internet est composé de milliers de systèmes autonomes (AS), dont le trafic inter-domaine est acheminé avec le protocole BGP (Border Gateway Protocol). En raison de la distribution naturelle et le manque d'autorisation dans BGP, un AS peut annoncer des chemins ou préfixes illégitimes appartenant à d'autres AS, c'est-à-dire, détourner leurs préfixes.

Le contrôleur ONOS propose une application « ARTEMIS » pour les administrateurs réseau, qui permet de détecter en temps réel et d'atténuer automatiquement les incidents de détournement de préfixe contre les préfixes sous leur contrôle administratif en utilisant l'autosurveillance au niveau du AS.

Mots clés : AS, BGP, ONOS, ARTEMIS, Distribution, Préfixe

Abstract

The Internet is made up of thousands of Autonomous Systems (AS), whose inter-domain traffic is routed using the BGP (Border Gateway Protocol). Due to natural distribution and lack of authorization in BGP, an AS can advertise illegitimate paths or prefixes belonging to other AS, that is, prefix hijacking.

The ONOS controller offer an application « ARTEMIS » for network administrators, that allows them to detect in real-time and automatically mitigate prefix hijacking incidents against prefixes under their administrative control, by employing self-monitoring on the AS level.

Key word: AS, BGP, ONOS, ARTEMIS, distribution, prefix, hijacking