الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Décmocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Nº Réf : ……

Centre Universitaire

Abdelhafid Boussouf Mila

Institut des Sciences et de la Technologie **Département de Mathématiques et Informatique**

Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique

Spécialité : Sciences et Technologies de l'Information et de la Communication

« IPv6 Traffic-Controller » : Un assistant pour le contrôle du trafic IPv6 dans un réseau SDN

Présenté par : **Laouar Aymen**

Bouaziz Anouar

Soutenu devant le jury :

Président : **Mr BESSOUF Hakim** Grade : **M.A.A**

Examinateur : **Mr SELAMANE Samir** Grade : **M.A.A**

Encadreur : **Mr BENCHEIKH LEHOCINE Madjed** Grade : **M.C.B**

Année Universitaire : 2021/2022

Remerciements

La louange est à Allah de par la grâce de qui se réalisent les bonnes choses, avec l'aide et le succès d'Allah, ce travail a été achevé.

Tout d'abord, nous tenons à remercier notre encadreur de mémoire **Mr BENCHIEKH EL HOCINE MADJED**, pour sa patience, sa disponibilité et ses précieux conseils, qui nous ont aidé tout au long de ce cheminement.

Nous tenons à remercier toutes les personnes qui ont contribué à la réussite de notre projet et qui nous ont aidés lors de la rédaction de ce mémoire.

Enfin, un grand merci à tous nos professeurs du centre universitaire **ABDELHAFID BOUSSOUF - MILA**, pour tout le savoir qu'ils nous ont inculqué.

Dédicace

Tout d'abord, je voudrais remercier ALLAH le tout puissant et miséricordieux, qui nous a aidé et qui nous a donné la force, le courage et la patience pour accomplir ce travail.

C'est avec un immense plaisir et une réelle joie de fierté que je dédie ce travail :

Á ma chère maman qui m'a soutenu et encouragé durant ces années d'études, pour ses précieux conseils, excellent soutien et sa patience.

Á mon cher père pour son travail acharné, pour son aide et son soutien tout au long de ces années;

Á mon frère ABDELWAHAB et ma sœur CHAIMA;

Á mes oncles et tantes : NAIMA, AHLAM, ADEL, ABDELHADI, YASSIN, FARIDA, FATIMA, ZOUBIR;

Á mes amis et à tous ceux qui m'ont aidé à terminer ce travail

Merci.

Aymen

Dédicace

Au Début et avant tout, je veux remercier le dieu qui à permet le courage à faire et finir ce modeste travail.

Je dédie ce modeste travail à ma honorable et aimable mère

A mon cher père

A mes chères sœurs

A tous les amis que j'ai connu jusqu'à maintenant. Merci pour leurs amours.

Et à tous mes collègues du lycée ainsi que les étudiants du STIC 2

À la fin, nous remercions tous ceux qui ont aidé de prés ou de loin à réaliser notre travail.

Anouar

Résumé

La digitalisation est devenue une tendance majeure qui se développe dans tous les domaines, et qui déclenche le besoin des services personnalisables pour les entreprises, les institutions et les utilisateurs finaux. Cette avancée technologique nécessite des développements dans le domaine de la gestion des réseaux, dans le but d'accélérer la mise en service des infrastructures adoptant des technologies (tel que IoT, IoE,…), des politiques de sécurité avancée, des protocoles de routages intelligents ou même des protocoles de communication tel que IPv6.

La technologie du réseau défini par logiciel (Software Defined networking ou SDN) a été conceptualisée pour simplifier la gestion du réseau, offrant la capacité de centraliser le contrôle du trafic ainsi que de nombreuses autres capacités qui ne sont pas disponible dans les réseaux traditionnels. Au cours des dernières années, la technologie SDN a attiré l'attention sur son utilité pour résoudre de nombreux problèmes de gestion liés au nombre croissant d'appareils connectés dans le monde, ce qui augmente les coûts d'administration en matière de ressources financières et humaines.

Dans le cadre de ce projet, nous avons développé une application qui permet de simplifier la spécification des politiques, relatives au contrôle du trafic IPv6 au niveau d'une infrastructure SDN utilisant le contrôleur RYU. Notre solution, « IPv6 Traffic-Controller », assure l'assistance des administrateurs dans l'objectif de personnaliser le comportement de leurs réseaux. Ceci est garanti à travers une ergonomie étudiée visant à minimiser les erreurs humaines tout en ayant une meilleure visibilité du flux, autorisé ou bien interdit, circulant dans le réseau.

Abstract

Digitalization has become a major trend that is growing in all areas, triggering the need for customizable services for companies, institutions and end users. This technological advance requires developments in the field of network management, with the aim of accelerating the provisioning of infrastructures adopting various technologies (such as IoT, IoE,….), advanced security policies, intelligent routing protocols or even communication protocols such as IPv6.

Software Defined Networking (SDN) technology was conceptualized to simplify network management, providing the ability to centralize traffic control as well as many other capabilities not available in traditional networks. In recent years, SDN technology has drawn attention to its usefulness in solving many management problems related to the growing number of connected devices in the world, which increases the costs of administration in terms of financial and human resources.

In this project, we have developed an application that simplifies the specification of policies relating to IPv6 traffic control in an SDN infrastructure using the RYU controller. Our solution, « IPv6 Traffic-Controller », assists network administrators in personalizing the behavior of their networks. This is guaranteed through a studied interface that aims to minimize human errors while having better visibility of the flow, authorized or denied, circulating in the network.

ملخص

لاقت الرقمنة رواجا كبيرا ينمو في جميع المجالات مما أدى الى الحاجة الى خدمات قابلة للتخصيص لفائدة الشركات والمؤسسات والمستخدمين. يتطلب استمرار هذا التقدم التكنولوجي التطوير في مجال ادارة الشبكات الالكترونية، بهدف

تسريع تشغيل البنى التحتية التي تعتمد على تقنيات مختلفة (مثل تقنيتي انترنت الاشياء وانترنت كل شيئ)، وسياسات الامان المتقدمة، وبروتوكولات التوجيه المتقدمة، وبروتوكولات التوجيه الذكية او حتى بروتوكولات الاتصال مثل IPv6.

تم تصميم تقنية الشبكات المعرفة بالبرمجيات (SDN) لتبسيط عمليات إدارة الشبكات ، مما يوفر القدرة على التحكم المركزي في حركة مرور البيانات بالإضافة إلى العديد من الإمكانات الأخرى غير المتوفرة في الشبكات التقليدية. وفي السنوات الأخيرة لفتت تقنية SDN الانتباه إلى فائدتها في حل العديد من مشاكل الإدارة المتعلقة بالتزايد المستمر للأجهزة المتصلة بالشبكات في العالم ، مما يزيد من تكاليف الإدارة من حيث الموارد المالية والبشرية.

في إطار هذا المشروع ، قمنا بتطوير تطبيق يبسط تعيين السياسات المتعلقة بالتحكم في حركة مرور IPv6 في البنية التحتية SDN باستخدام وحدة تحكم RYU. يساعد النظام الذي اطلقنا عليه اسم « IPv6 Traffic-Controller » مديري الشبكات في تخصيص سلوك شبكاتهم. يتم ذلك من خلال ضمان بيئة عمل مدروسة تهدف إلى تقليل الأخطاء البشرية مع الحصول على رؤية أفضل لتدفق البيانات المرخص بها أو المرفوضة المتداولة في الشبكة.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

Liste des abréviations

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IETF Internet Engineering Task Force

FAI Fournisseur d'Accès à Internet

IoT Internet of Things

IoE Internet of Everything

AWS Amazon Web Services

IBM International Business Machines

5G 5th generation mobile network

PIB Produit Interieur Brut

TOS Type Of Service

DSCP Differentiated Services Code Point

TTL Time To Live

ECN Explicit Congestion Notification

RFC Request for Comments

RIPng Routing Information Protocol next generation

DHCP Dynamic Host Configuration Protocol

EIGRPv6 Enhanced Interior Gateway Routing Protocol version 6

TCP Transmission Control Protocol

SDN Software Defined Networking

ONF Open Networking Foundation

CSP Communications Service Providers

SD-WAN Software-Defined Wide Area Network

QoS Quality of Service

JSON JavaScript Object Notation

ICMPv6 Internet Control Message Protocol vesion 6

OSPFv3 Open Shortest Path First version 3

REST REpresentational State Transfer

API Application Programming Interface

JEE Java Platform Enterprise Edition

JDK Java Développent Kit

UML Unified Modeling Language

OMG Object Management Group

UP Unified Process

EDI Electronic Data Interchange

SGBDR Système de gestion de base de données relationnelle

INTRODUCTION GÉNÉRALE

1. Contexte

L'influence mondiale du réseau Internet est bien reconnue avec un taux de pénétration phénoménal. La façon dont les affaires des entreprises et des sociétés sont conduites dans cette ère numérique a changé en raison du nombre important des personnes, dépassant le tiers de la population, connectées à Internet. Les différentes organisations utilisent ce réseau mondial pour commercialiser et promouvoir des produits et des services.
Cependant, la demande croissante des entreprises, des organisations et des fournisseurs d'accès à Internet pour les adresses IPv4 (Internet Protocol version 4) publique a provoqué un épuisement du nombre d'adresses disponibles, ce qui a conduit au développement du protocole IPv6 (Internet Protocol version 6) qui est destiné à remplacer graduellement IPv4. Le développement de ce protocole a encouragé les chercheurs et les vendeurs à l'implémenter dans les nouvelles technologies émergentes. Le nouveau paradigme Software-Defined Networking (SDN) est l'un de ces technologies.
Le paradigme SDN offre la capacité de créer des réseaux programmables, ce qui permet de personnaliser le comportement des réseaux. Pratiquement, les administrateurs ont la possibilité de gérer leurs réseaux à travers des applications de manière logiquement centralisée, en ayant une vision globale de la topologie, quel que soit sa taille.. La mise en œuvre de la technologie IPv6 dans l'infrastructure SDN offre de grandes opportunités pour les entreprises modernes en matière d'évolutivité et de d'adaptabilité.

2. Problématique

Bien que la mise en œuvre d'IPv6 dans le réseau SDN apporte de grands avantages pour les entreprises modernes, la gestion et la mise en service de tels réseaux restent complexes, surtout en considérant la taille du système d'adressage d'IPv6 qui est de l'ordre de 2^{128} adresses. Cela rend le contrôle du trafic coûteux, lent et sujet aux erreurs.

3. Objectif

L'objectif de ce projet est de développer la solution « IPv6 Traffic-Controller » dont le rôle est d'assister les administrateurs dans la tâche du contrôle de trafic IPv6 au niveau des réseaux SDN. Cette solution vise à simplifier la planification hiérarchique du réseau dans le but d'assurer la fiabilité des différentes opérations relatives aux contrôle du trafic IPv6. L'avantage principal de ce projet est

la réduction du coût de la gestion en minimisant le temps de déploiement et en réduisant l'interaction humaine qui est toujours sujettes aux erreurs.

4. Organisation du mémoire

Ce mémoire est organisé en trois chapitres comme suit :

— Le premier chapitre commence par introduire le protocole IPv6 et ses apports dans les technologies de pointe. Ensuite, il présente les concepts de base relatifs à ce protocole.

— Le deuxième chapitre expose la motivation de ce projet. Il commence par introduire les principes fondamentaux du paradigme SDN. Ensuite, il présente, à travers une simulation, la problématique liée à la spécification manuelle des politiques relatives au trafic IPv6 dans un réseau SDN utilisant le contrôleur RYU. La fin de ce chapitre est consacrée aux détails de la solution proposée qui offre à l'administrateur un outil très puissant pour un déploiement rapide, maitrisable et contrôlé de son réseau IPv6.

— Le troisième chapitre présente la conception de notre système, proposé dans le cadre de ce projet, suivi par une présentation de l'environnement de développement ainsi que les outils exploités pour la réalisation de ce travail.

CHAPITRE 1

IPV6

1.1 Introduction

Dans le monde des réseaux, un protocole est un ensemble de règles qui spécifient comment un paquet de bits sera interprété et comment un nœud doit réagir aux datagrammes qu'il reçoit. Le protocole IPv4 est l'un des piliers du réseau mondial Internet. La demande croissante en adresses pour les nouvelles applications, les équipements mobiles et les équipements connectés en permanence à conduit au développement d'une nouvelle version d'IP. Les travaux sur IPv6 ont été menés par l'IETF (Internet Engineering Task Force) [1] pour résoudre les problèmes relatifs à d'épuisement des adresses IPv4.
Dans ce chapitre nous allons souligner l'importance de la technologie IPv6 et le grand rôle qu'elle joue dans l'avancement des technologies modernes et émergentes. Ensuite, nous allons présenter les bases relatives à ce protocole.

1.2 Le Protocol IPV6

IPv6 est un protocole de la couche Internet dont le rôle est principalement est l'interconnexion des réseaux à commutation de paquets. Il fournit une transmission de datagrammes de bout en bout sur plusieurs réseaux IP, en respectant étroitement les principes de conception développés dans la version précédente du protocole « Internet Protocol Version 4 » (IPv4).
En plus d'offrir plus d'adresses, IPv6 implémente également des fonctionnalités non présentes dans IPv4. Il simplifie les aspects de la configuration des adresses ainsi que les annonces de routeur lors du changement du fournisseur de connectivité réseau. Il simplifie également le traitement des paquets dans les routeurs en confiant la responsabilité de la fragmentation des paquets aux points d'extrémité. La taille du sous-réseau IPv6 est normalisée en fixant la taille de la partie identifiant d'hôte à 64 bits.

1.3 Pourquoi IPv6 ?

Au cours des dernières années, la technologie IPv6 est de plus en plus utilisée à la place d'IPv4. C'est principalement parce que les FAI (Fournisseur d'Accès à Internet) ne peuvent pas fournir suffisamment d'adresses IP aux appareils mobiles ainsi qu'aux objets connectés, constituant les solutions IoT (Internet des objets), en constante augmentation.
La plupart des déploiements IPv6 sont limités aux réseaux câblés. Cependant, l'augmentation rapide du nombre d'appareils mobiles connectés à Internet a obligé les entreprises de télécommunications

mobiles à déployer la technologie IPv6 sur leurs réseaux [2].
Ce mouvement vers la communication IPv6 encourage les chercheurs et ouvre la voie à des projets de développement dans divers domaines, en particulier ceux qui nécessitent ou incluent une connexion à un réseau local (LAN) ou mondial (WAN).

1.3.1 IPv6 dans l'Iot

Le domaine de l'IoT (Internet of Things) est encore en développement et de nombreuses études portent sur les méthodes de mise en œuvre de l'IoT dans lesquelles les auteurs utilisent le protocole IPv6 en raison de ses avantages par rapport à IPv4. Par exemple, dans [3], les auteurs étudient le « Fog Computing » et les plates-formes des systèmes embarqués comme des éléments constitutifs de l'IoT. Ils abordent diverses plates-formes et technologies, et ils prouvent que le protocole IPv6 est plus sécurisé qu'IPv4. Dans un autre travail de recherche [4], les auteurs ont mentionné que l'un des avantages les plus connus d'IPv6 dans le domaine de l'IoT est la configuration automatique de la communication et la découverte de ressources sans configuration.
Dans un rapport [5] publié en novembre 2020, les auteurs ont étudié la croissance mondiale d'IPv6 et ils ont mentionné que la plupart des nouveaux appareils IoT ont une adresse IPv6 de type Link-Local implémenté par défaut. Actuellement, les chercheurs étudient les vulnérabilités potentielles d'IPv6 dans les appareils IoT pour garantir une connexion sécurisée à internet.

1.3.2 IPv6 dans le Cloud Computing

De nombreux fournisseurs de services cloud ont commencé à prendre en charge les technologies IPv6 dans leurs réseaux et serveurs en raison des facteurs mentionnés précédemment. Nous citons certaines des entreprises les plus connues :

- **Amazon Web Services (AWS) :**

AWS est une filiale d'Amazon qui fournit des plates-formes de cloud computing et des API à la demande aux particuliers, aux entreprises et aux gouvernements. Les services d'AWS prennent en charge IPv6 depuis 2011 et, à partir de 2021, ils disposent des fonctionnalités purement IPv6 [6].

- **Microsoft Azure :**

C'est un service de cloud computing exploité par Microsoft pour la gestion des applications via des centres de données. L'entreprise a annoncé en avril 2020 que la prise en charge d'IPv6 au sein du

réseau virtuel Azure et sur Internet est généralement disponible dans le monde entier. Ils ont mentionné que la raison de cette mise à niveau de leurs services est la croissance des marchés mobiles ainsi que celui de l'IoT avec des applications basées sur Azure [7].

• Google cloud :

Depuis 2017, Google a commencé à ajouter le support IPv6 à ses services [8], et ils ont continué à étendre graduellement la prise en charge d'IPv6. Ce qui signifie un intérêt croissant pour l'adoption d'IPv6 dans autant de ses services que possible.

• Oracle cloud :

En avril 2021, Oracle cloud a annoncé avoir ajouté la prise en charge d'IPv6 sur son infrastructure cloud, offrant aux clients diverses fonctionnalités destinées aux appareils IoT, aux applications mobiles, aux sites Web et aux réseaux sur site [9].

1.3.3 Ipv6 dans la 5G

La 5G est le réseau mobile de 5ème génération. Il s'agit d'une nouvelle norme sans fil mondiale. Cette nouvelle technologie est censée offrir des vitesses de données de pointe multi-Gbps, une latence très faible, plus de fiabilité et une capacité réseau massive [10].

La 5G est utilisée dans trois principaux types de services connectés, notamment (1) le haut débit mobile, (2) les communications critiques et (3) l'IoT massif (cas d'utilisation étendus) [10]. Cela signifie qu'une quantité massive de nouveaux appareils pourront être connectés à Internet via le réseau 5G. Précisément, la 5G devrait prendre en charge jusqu'à 1 million d'appareils connectés par près d'un kilomètre carré, contre environ 2 000 appareils connectés par un kilométrage carré avec la 4G [11].

1.3.4 Impact économique du déploiement d'IPv6

Comme mentionné précédemment, IPv6 améliore le déploiement d'applications professionnelles avancées telles que la 5G, le cloud et l'IoT. Ils constituent également la base de la numérisation de toutes les futures industries.

Dans un livre blanc [12] publié en 2021 les auteurs ont expliqué en détail l'impact social et économique du déploiement de l'ipv6 dans le monde. Ils estimaient que la valeur globale potentielle créée à travers

plusieurs secteurs industriels activés par IPv6 pourraient atteindre 10,8 billions de dollars en 2030. Cette estimation très élevée s'explique par les avantages économiques et sociaux d'IPv6, notamment :

— Soutenir la croissance économique dans l'ère post-épidémique en soutenant la technologie à distance.
— Dynamiser la transformation de l'économie numérique.
— Stimuler l'innovation et l'entrepreneuriat.
— Promouvoir l'efficacité de l'IoT.
— Faciliter la collecte de données pour l'IA.

1.3.5 Déploiement IPv6 dans le monde

Dans le même livre blanc [12], les auteurs ont pris les statistiques de déploiement IPv6 dans des pays entre 2015 et 2020 et ils ont constaté que les pays précurseurs en matière d'adoption d'IPv6 ont un Produit Intérieur Brut (PIB) plus élevé que les pays débutants. Le PIB est le principal indicateur de la mesure de la production économique réalisée à l'intérieur d'un pays. La conclusion à laquelle les auteurs sont parvenus est que les actions gouvernementales ont des effets importants sur l'adoption globale de la technologie IPv6, et que les pays à faible croissance du déploiement devraient tirer parti des politiques associées au déploiement d'IPv6.

1.4 Les bases d'IPv6

1.4.1 En-tête IPv6

Le format de l'en-tête IPv6 est illustré dans la **Figure 1.1** :

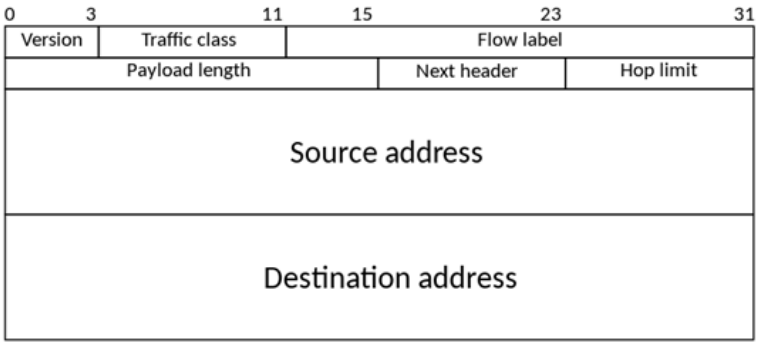

FIGURE 1.1 – Format de l'en-tête IPv6 [13].

La signification des champs est la suivante :

• **Version (4 bits) :** Fixée à la valeur de la version du protocole internet = 6.

• **Traffic Class (8 bits) :** Ce champ indique la classe ou la priorité du paquet IPv6. Il est similaire au champ Type Of Service (TOS) dans l'entête IPv4. Les six premiers bits du champ Classe de Trafic représentent le champ Differentiated Services Code Point (DSCP) [14], et les deux derniers bits sont utilisés pour signaler la congestion du réseau (Explicit Congestion Notification ou ECN) [15].

• **Flow Label (20 bits) :** Le champ Flow Label indique que ce paquet appartient à une séquence spécifique de paquets entre une source et une destination. L'utilisation du champ Flow Label permet une classification efficace des flux IPv6 [16].

• **Payload length (16 bits) :** Indique la longueur du reste du paquet qui suit l'en-tête IPv6, il inclut la longueur de tous les en-têtes d'extension [1].

• **Next Header (8 bits) :** Identifie le type de l'en-tête qui suit immédiatement l'en-tête IPv6 et utilise les mêmes valeurs que le champ Protocole dans l'entête IPv4.

• **Hop Limit (8 bits) :** Identique au champ TTL (Time To Live) dans l'entête IPv4. Il indique le nombre maximum des nœuds intermédiaires que le paquet IPv6 est autorisé à parcourir. Sa valeur est décrémentée de un par chaque nœud qui transmet le paquet. Une fois que le TTL atteint la valeur « 0 » le paquet sera rejeté. Ceci est utilisé pour éliminer les paquets bloqués dans une boucle infinie à cause d'une erreur de routage.

• **Source Address (128 bits) :** Adresse source.

• **Destination Address (128 bits) :** Adresse destination.

1.4.2 L'Adresse IPV6

Une adresse IPv6 a une longueur de 128 bits. Elle est composée de huit champs de 16 bits, chaque champ étant délimité par deux points et représenté par des chiffres hexadécimaux, contrairement à la notation décimale à points des adresses IPv4.
Un exemple d'une adresse IPv6 est le suivant : **2001 :0db8 :3c4d :0051 :4408 :0a40 :bde0 :0525**

Le terme non officiel pour une section de quatre valeurs hexadécimales est un 'Hextet' [17]. Par exemple : le champ «2001» est un 'Hextet', similaire au terme 'Octet' utilisé dans l'adressage IPv4.

1.4.2.1 Notation du préfixe dans une adresse IPv6

Dans le protocole IPv4, le préfixe (ou la partie réseau) de l'adresse peut être identifié par un masque de sous-réseau [18]. Par exemple, **192.168.1.50 255.255.255.0** indique que la partie réseau de l'adresse IPv4 correspond à : « **192.168.1** ». Le masque de réseau décimal à point **255.255.255.0** peut également être représenté par une notation CIDR (Classless Inter-domain Routing) sous la forme « **/24** » [19], indiquant que les 24 bits les plus à gauche représente la partie préfixe.
Le préfixe d'une adresse IPv6 peut être représenté d'une manière similaire à la notation CIDR de l'IPv4, comme le montre la **Figure 1.2** :

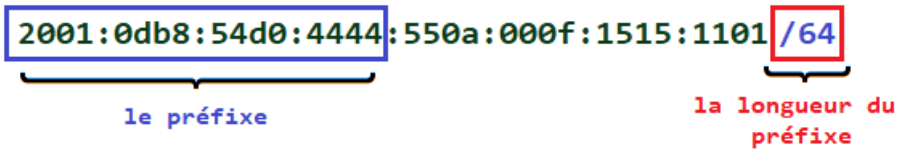

FIGURE 1.2 – Exemple d'une adresse IPv6.

1.4.2.2 Abréviation dans l'adressage IPV6

Prenons l'exemple suivant d'une adresse IPV6 :

2001 :0db8 :0000 :0000 :3c4d :0000 :0000 :0525

La RFC 5952 fournit deux règles utiles pour réduire la notation impliquée dans le format :
o La première règle consiste à omettre les 0 de tête :
Une façon de raccourcir les adresses IPv6 consiste à omettre les 0 de début dans n'importe quel hextet. Cette règle s'applique uniquement aux 0 de début et non aux 0 de fin [20].
Donc, à partir de l'adresse IPv6 précédente, les 0 de tête sont marqués en gras :

2001 :**0db8** :**000**0 :**000**0 :3c40 :**000**0 :**000**0 :**0525**

Après l'application de la première règle, on obtient :

2001 :db8 :0 :0 :3c40 :0 :0 :525

o La deuxième règle consiste à omettre les hextets contenant uniquement des zéros :

Cette règle consiste à utiliser un double deux-points (: :) pour représenter n'importe quelle chaîne unique et contiguë de deux hextets ou plus composés uniquement de 0. Nous pouvons utiliser (: :) une seule fois dans une adresse IPv6.

En utilisant l'exemple précédent, nous aurons deux représentations possibles (voir les deux **Figures 1.3** et **1.4**) :

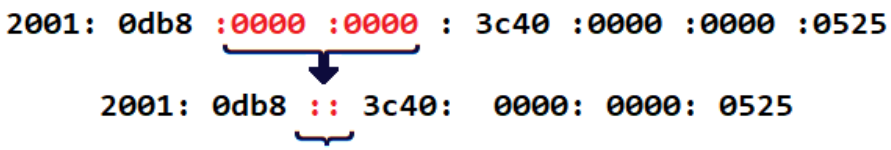

FIGURE 1.3 – Abréviation de l'adresse IPv6 – Cas N°1.

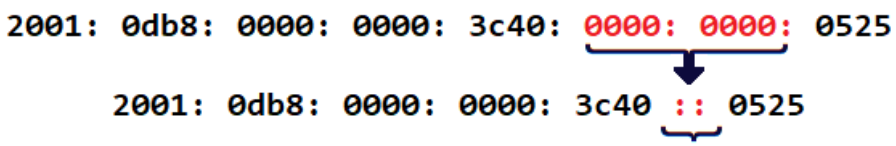

FIGURE 1.4 – Abréviation de l'adresse IPv6 – Cas N°2.

Les deux abréviations sont correctes, mais selon [20], il est recommandé d'abréger la première séquence des bits à zéro.

1.4.3 Types des adresses IPv6

Dans l'IPv4, les adresses sont classées en trois types de base : Unicast, Multicast et Broadcast [21]. Avec IPv6, le type Multicast remplace le type Broadcast [22]. De plus, le type Anycast a été ajouté à IPV6 pour être explicitement prise en charge, contrairement à IPv4 où il n'est pas officiellement pris en charge [23].

La **Figure 1.5** montre les trois types des adresses IPv6 ainsi que ce qui relève de chaque type.

1.4.3.1 Unicast

Une adresse Unicast identifie de manière unique une interface sur un périphérique IPv6. Un paquet envoyé à une adresse Unicast est reçu par l'interface configurée avec cette adresse. Ci-dessous, nous mentionnons les types des adresses Unicast :

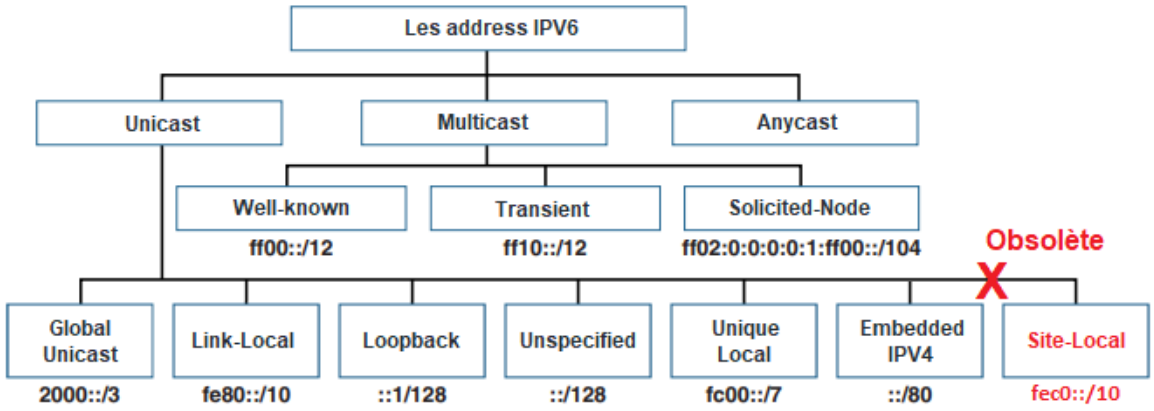

FIGURE 1.5 – Types des adresses IPv6.

Adresse Global Unicast :

Le format général des adresses IPv6 Global Unicast est comme indiqué dans la **Figure 1.6** [23] :

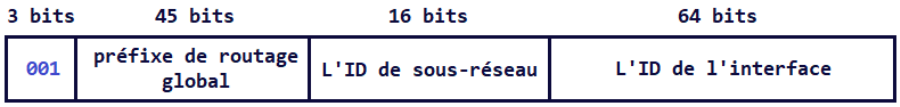

FIGURE 1.6 – Le format général de l'adresse IPv6 Global Unicast.

Une adresse Global Unicast commence par les trois bits « 001 ». Le préfixe de routage global est une valeur, codée sur 45 bits, qui permet d'identifier un site (un groupe de sous-réseaux). Le champ suivant, codé sur 16 bits, représente l'ID du sous-réseau. L'adresse Global Unicast est globalement unique et elle est routable sur Internet (l'équivalent d'une adresse IPv4 publique).

Adresse Link-Local Unicast :

Les adresses Link-Local sont limitées au lien et ne peuvent pas être routées au-delà du sous-réseau local [23]. Le bloc des adresses qui a été réservé pour les adresses Link-Local est « fe80 : :/10 » (voir la **Figure 1.7**).

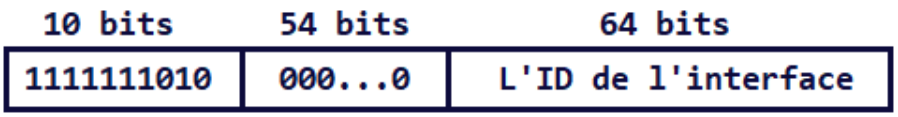

FIGURE 1.7 – le format de base des adresses Link-Local

Un périphérique IPv6 doit obligatoirement avoir une adresse Link-Local qui est générée automatiquement par le système d'exploitation. Ce type d'adresse est utilisé pour les mécanismes d'auto-configuration ainsi que pour la découverte de voisins [24].

Adresse Loopback :

L'adresse Loopback IPv6 est « : :1 », c'est l'équivalent du bloc d'adresses IPv4 127.0.0.0/8. Les paquets IPv6 avec une adresse de destination Loopback ne doivent pas être envoyés en dehors du nœud d'origine [23].

Adresse Unspecified

L'adresse non spécifiée « 0 :0 :0 :0 :0 :0 :0 :0 » indique l'absence de l'adresse IPv6. Un exemple de son utilisation est dans le champ adresse source d'un paquet IPv6 DHCP (Dynamic Host Configuration Protocol) request.

Adresse Unique Local

Une adresse locale unique IPv6 (voir la **Figure 1.8)** a été conçue pour être unique dans les communications locales. Ces adresses sont routables à l'intérieur d'une zone plus restreinte, telle qu'un site. Ils peuvent également être routés entre un ensemble limité de sites, mais ils ne sont pas routables sur Internet [25].

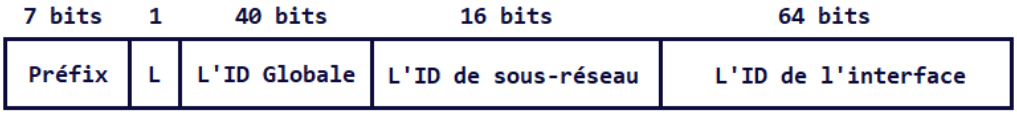

FIGURE 1.8 – le format de base des adresses Unique Local [26].

L'adresse Unique Local est composée des champs suivant [25] :

- Le champ « préfixe », codé sur 07 bits, sert à identifier les adresses IPv6 Unique
- Le champ « L » est toujours mis à 1.
- L'« ID global » est utilisé pour identifier un préfixe globalement unique.

Site-Local (obsolète) :

Le préfixe des adresses Site-Local était « fec0 : :/10 ». Ce type d'adresse est obsolète dans la RFC 3879 [24]. Il a été remplacé par l'adresse IPv6 locale unique.

Embedded

Les adresses Embedded permettent de transporter des adresses IPv4. Il existe deux types de ces adresses :

• **IPv4-Compatible IPv6 Address :** L'adresse IPv6 de ce type a été définie pour faciliter la transition vers un réseau entièrement IPv6 [23].

• **IPv4-Mapped IPv6 Address :** Ce type d'adresse est utilisé pour représenter les adresses des nœuds IPv4 sous forme d'adresses IPv6. Un nœud IPv6 peut utiliser cette adresse pour envoyer un paquet à un nœud IPv4 uniquement [23].

1.4.3.2 Multicast

Une adresse IPv6 Multicast est un identifiant pour un groupe d'interfaces (généralement sur différents nœuds). Une interface peut appartenir à n'importe quel nombre de groupes Multicast. Les adresses multicast ont le format suivant (voir la **Figure 1.9**) [23] :

FIGURE 1.9 – le format de base des adresses Multicast.

La suite binaire « 11111111 » au début de l'adresse permet de l'identifier comme étant une adresse multicast. Le champ « flags » est un ensemble de 4 drapeaux, chacun d'eux étant codé sur un bit [23]. Le champ « Scope » comprend une valeur, codé sur 4 bits, permettant de spécifier l'étendue du groupe Multicast. Par exemple, la valeur « 5 » signifie que l'adresse est limitée à une étendue Site-Local [23].

1.4.3.3 Anycast

Une adresse IPv6 Anycast est une adresse IPv6 qui est attribuée à une ou plusieurs interfaces réseau appartenant généralement à des nœuds différents. Un paquet envoyé à une adresse Anycast est acheminé vers le nœud le plus proche ayant cette adresse, selon la mesure de distance des protocoles de routage [23].

1.4.4 Routage IPv6

Le routage est le processus consistant à choisir le meilleur chemin pour transmettre un paquet à partir d'une source vers une destination via une série de routeurs. Le routeur est l'équipement qui

prend cette décision en consultant sa table de routage [26].
Il existe deux méthodes pour remplir la table de routage : le routage statique et le routage dynamique [27].

1.4.4.1 Routage statique

L'administrateur alimente manuellement la table de routage en introduisant les routes nécessaires à l'acheminement du trafic réseau. Cette méthode de routage est mieux utilisée dans les réseaux de petite taille.

1.4.4.2 Routage dynamique

Lorsqu'on parle du routage dynamique, les routeurs utilisent des protocoles de routage spécifiques pour remplir la table de routage. Un routeur peut exécuter plusieurs protocoles de routage pour la redondance et choisit la meilleure route apprise par le protocole ayant la priorité la plus élevée.
Il existe trois types de protocoles de routage : (1) à vecteur de distance, (2) à état de lien et (3) hybride.

Vecteur de distance (Distance Vector) :
Les protocoles de routage à vecteur de distance sont utilisés pour trouver le meilleur chemin vers un réseau distant en calculant la distance représentée par le nombre de sauts. Un saut est compté lorsqu'un paquet traverse un routeur. La route ayant la distance la plus basse vers un réseau est déterminée comme étant le meilleur chemin. L'algorithme utilisé par les protocoles de vecteurs de distance pour calculer le meilleur chemin est " Bellman Ford". RIPng (Routing Information Protocol next generation) est un protocole à vecteur de distance spécifié pour prendre en charge IPv6.

État du lien (Link State) :
Les protocoles de routage à état des liens sont utilisés pour trouver le meilleur chemin en fonction du coût. Donc, l'itinéraire avec le coût le plus bas est déterminé comme étant le meilleur chemin. Les protocoles de routage à état des liens utilisent l'algorithme "Dijkstra" qui est plus rapide et plus fiable que l'algorithme utilisé par les protocoles à vecteur de distance. Il nécessite moins de bande passante et ne fait face à aucun risque de boucles de routage. OSPFv3 (Open Shortest Path First version 3) prend en charge IPv6.

Hybride :
Les protocoles de routage hybrides utilisent à la fois des fonctionnalités de vecteur de distance et

d'état de liaison. EIGRPv6 (Enhanced Interior Gateway Routing Protocol version 6) est l'exemple des protocoles de routage hybrides.

1.5 Conclusion

Dans ce chapitre, nous avons bien exposé l'importance du protocole IPv6 dans les réseaux modernes, et ses apports dans les différentes technologies émergentes. Nous avons aussi présenté les bases relatives à l'adressage IPv6 ainsi que les techniques de routage qui supportent ce protocole.
Dans le chapitre suivant, nous allons présenter les réseaux de type Software Defined Netwoking (SDN) dans le but d'orienter cette étude vers le contrôle du trafic IPv6 au niveau d'une infrastructure SDN.

CHAPITRE 2

SOFTWARE DEFINED NETWORKING : VERS UN CONTRÔLE FIABLE DU TRAFIC IPV6

2.1 Introduction

Le Software Defined Networking (SDN) modifie radicalement l'architecture du réseau en dissociant le contrôle des dispositifs de transfert sous-jacents. Ce changement architectural permet une gestion centralisée et une reprogrammabilité des réseaux, donnant aux administrateurs la possibilité d'innover et de réaliser des avancées importantes dans la couche de contrôle. Cependant, l'approche manuelle dans l'administration des réseaux et ses défauts restent un problème même pour la technologie SDN.
Dans ce chapitre, nous allons détailler les aspects et les caractéristiques de la technologie SDN, ainsi que ses avantages et ses domaines d'application. Puis, nous allons exposer la problématique, traitée par ce projet, en simulant la solution de RYU pour le contrôle du trafic IPv6 dans un réseau SDN. Enfin, nous allons introduire notre solution qui simplifie largement la tache de gestion en assistant l'administrateur dans l'ensemble des opérations relatives au contrôle du trafic.

2.2 La technologie SDN

2.2.1 Définition du SDN

Le terme Software Defined Networking peut être traduit par « Réseau Défini par Logiciel ». C'est un nouveau paradigme dont l'avantage crucial est la programmabilité des différentes fonctionnalités réseaux.

L'ONF (Open Networking Foundation, 2013) définit le SDN comme une architecture réseau où le plan de contrôle est totalement découplé du plan de données. La logique du SDN unifie les plans de contrôle de plusieurs périphériques dans un seul software externe appelé « Contrôleur ». Cet élément dispose d'une vue globale du réseau permettant de gérer l'infrastructure via des interfaces de communications appelées APIs.

2.2.2 imporatnce du SDN

L'objectif de des améliorations apportées par SDN est de simplifier l'administration des réseaux et, à l'instar de ce que la virtualisation des serveurs a accompli, d'augmenter la flexibilité de la manière dont les applications utilisent les ressources du réseau.
Les atouts qui semblent d'une importance primordiale dans l'environnement SDN sont les suivants :

Gestion de réseau plus efficace : Le SDN offre une visibilité en temps réel sur les performances du réseau. Cette visibilité permet d'optimiser la qualité du réseau et de piloter son efficacité [28].

Économie des coûts : Dans les réseaux traditionnels, le moyen le plus efficace pour renforcer la disponibilité du réseau était la redondance. Ceci s'accompagnait bien sûr de plus d'équipements, plus de circuits et par conséquent des coûts supplémentaires. Étant donné que le SDN offre la capacité, en temps réel, de rediriger automatiquement le trafic ou de mettre en place de nouvelles fonctions et routes, les administrateurs peuvent augmenter la disponibilité sans ajouter de nouveau matériel ni d'augmenter les coûts [28].

Évolutivité plus rapide : L'SDN est conçu pour doter les administrateurs de la capacité opérationnelle d'adapter leurs besoins fonctionnelles très rapidement même avec des larges topologies [28].

2.2.3 Architecture

Le SDN présente une architecture réseau composée de trois couches communiquant entre elles par le biais d'interfaces APIs (comme l'illustre la **Figure 2.1**) :

• La couche infrastructure : Cette couche est constituée des commutateurs physiques du réseau. Autrement dit, les switchs SDN responsables de l'acheminement du trafic.

• La couche de contrôle : C'est le cerveau du réseau qui englobe la plupart des opérations de calcul.

• La couche application : Cette couche contient des programmes, qui transmettent des instructions spécifiques au contrôleur SDN. Elle permet aux administrateurs de configurer, gérer, sécuriser et optimiser les ressources du réseau via ces applications.

2.2.4 Le protocole OpenFlow

• Définition :

OpenFlow est un protocole qui permet la communication entre le contrôleur SDN et les commutateurs qui supportent OpenFlow. Il opère au cours d'une session TCP (Transmission Control Protocol) via le port 6633 du serveur contrôleur.

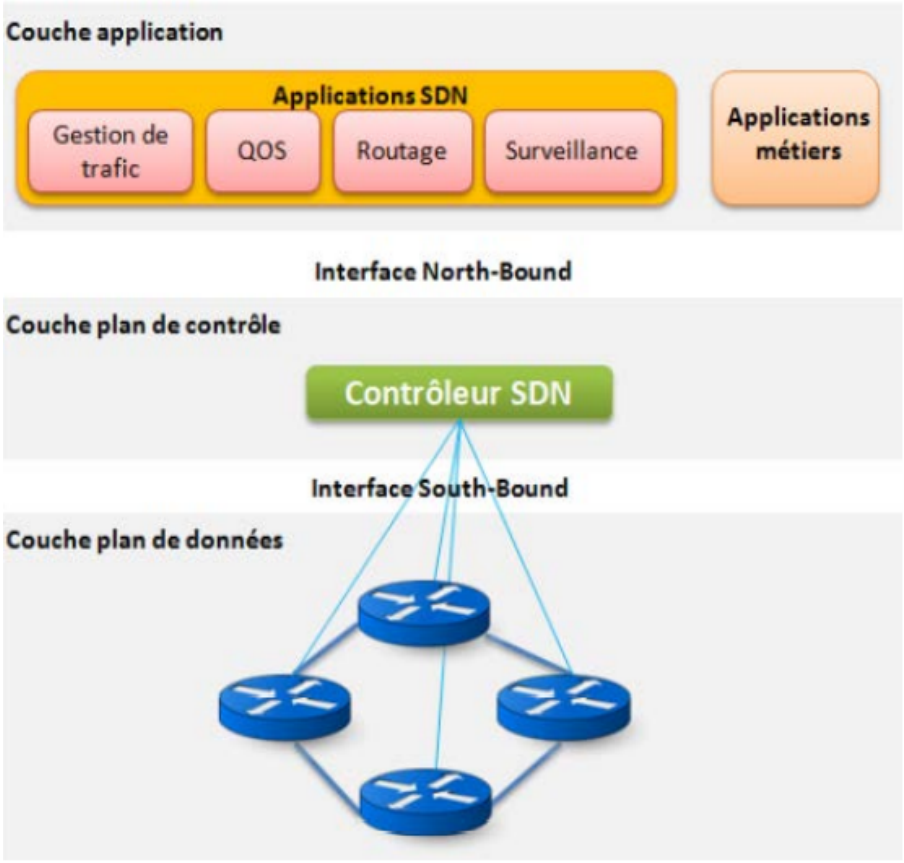

FIGURE 2.1 – Architecture du réseau SDN

Le comportement du switch OpenFlow est déterminé par une ou plusieurs tables de flux (flow table). Chaque table contient un ensemble de flux ou de règles qui s'appliquent au trafic reçu [29].
OpenFlow a été initié comme un projet à l'université de Stanford lorsqu'un groupe de chercheurs exploraient la manière de tester de nouveaux protocoles dans le monde IP (en créant un réseau expérimental confondu avec le réseau de production) mais sans arrêter le trafic du réseau de production lors des tests. C'est dans cet environnement que les chercheurs à Stanford ont trouvé un moyen de séparer le trafic de recherche du trafic du réseau de production qui utilise le même réseau IP [30].

• Architecture du protocole Openflow

L'essence du protocole Openflow consiste en un ensemble de messages qui transitent entre le contrôleur et le switch dans les deux sens . Ce sont ces messages qui permettent au contrôleur de gérer les switchs ainsi que le trafic des utilisateurs [31].

• Tables de flux

Chaque table de flux contient un ensemble d'entrées qui présentent les règles d'acheminement des paquets (comme l'illustre la **Figure 2.2**). Une entrée de flux est constituée de :

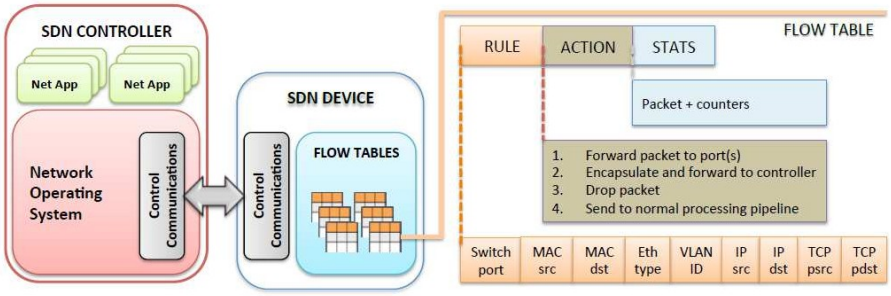

FIGURE 2.2 – Table de flux dans les switchs SDN [32].

Match fields : Une correspondance (match) consiste en un ou plusieurs contraintes concernant les champs des entêtes. Ces contraintes doivent toutes être respectées pour satisfaire la correspondance.

Action : Chaque entrée est associée à zéro ou plusieurs actions qui spécifient comment le commutateur gère les paquets correspondants. Si aucune action n'est entreprise, le paquet sera supprimé. La liste d'actions contenue dans les entrées doivent être traitées dans l'ordre spécifié.

Counters : Certains compteurs sur les paquets.

2.2.5 Les contrôleurs SDN

Le rôle du plan de contrôle est de gérer les équipements de l'infrastructure et de les relier avec les applications. Ce plan est composé d'un ou de plusieurs contrôleurs et il est considéré comme le système d'exploitation du réseau [33].

Le contrôleur SDN permet d'implémenter rapidement un changement sur le réseau en traduisant une demande globale (par exemple : prioriser l'application X) en une suite d'opérations sur les équipements réseau. Le contrôleur communique avec les équipements via une ou plusieurs API dites « Southbound » ou API sud [35].

Exemples des contrôleurs SDN

NOX : Un contrôleur initialement développé par Nicira Networks. Il sert de plate-forme de contrôle de réseau écrite en C++. Cette plateforme fournit une interface de programmation de haut niveau pour la gestion et le développement d'applications réseau [36].

POX : Un contrôleur basé sur NOX en langage python, dont le but est d'améliorer les performances du contrôleur original NOX [34].

Beacon : Développé en java par l'université de Stanford, il est aussi basé sur les technologies de multithreads. Son architecture modulaire permet au gestionnaire d'exécuter uniquement les services désirés [34].

Floodlight : C'est une variante de Beacon, caractérisée par sa simplicité et sa performance. Ce contrôleur a été testé avec les commutateurs OpenFlow physiques et virtuels. Il est aujourd'hui supporté par une large communauté de développeurs, comprenant des industriels comme Intel, Cisco, HP, Big switch et IBM [34].

Opendaylight : OpenDaylight est un projet de la fondation linux pris en charge par l'industrie. C'est un framework de source ouverte (open-source). Comme Floodlight, il peut également être considéré comme une solution complète [37].

RYU : RYU est un framework SDN qui fournit des composants logiciels avec une API bien définie (voir la **Figure 2.3**). Il permet aux développeurs de créer facilement de nouvelles applications pour la gestion et le contrôle du réseau. Ryu est basé sur Python et supporte la majorité des versions d'OpenFlow [38].

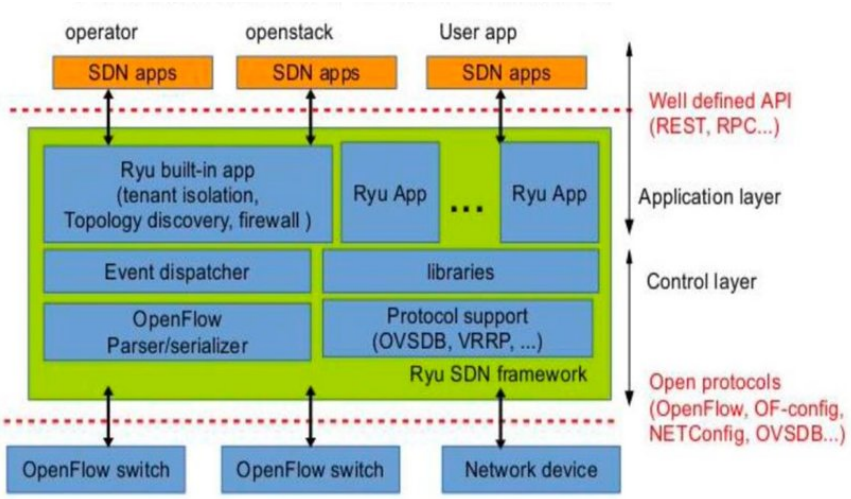

FIGURE 2.3 – L'architecture du contrôleur Ryu [39].

2.2.6 Les interfaces de communication

• **L'interface sud (Southbound)** Southbound est l'une des composantes les plus critiques dans l'architecture SDN, c'est une interface de communication permettant au contrôleur d'interagir avec les nœuds de la couche d'infrastructure.

• **L'interface nord (Northbound)** Les interfaces Nord servent à programmer les éléments de la transmission en exploitant l'abstraction du réseau fourni par le plan de contrôle. Il n'existe aucun standard intervenant entre la couche de contrôle et celle d'application du côté d'ONF ou d'autres organisations. Selon l'ONF, plusieurs niveaux d'abstraction et différents cas d'utilisation peuvent être caractérisés, ce qui signifie qu'il peut y avoir plusieurs interfaces Nord pour servir tous les cas d'utilisation. Parmi les propositions les plus réputées nous citons l'API basée sur REST (REpresentational State Transfer) [34].

2.2.7 Impact du SDN sur les nouvelles technologies

Dans un article de presse de businesswire [40], les estimations montrent que l'adoption de la technologie SDN augmente dans le monde entier et que les principaux moteurs de croissance du marché sont l'investissement des fournisseurs de services de communications (CSP) dans la technologie SDN pour automatiser l'infrastructure réseau.
Cette croissance a encouragé les scientifiques et les chercheurs du monde entier à découvrir les nombreuses façons dont la technologie SDN peut être utilisée pour innover dans divers domaines, et cela

inclut : les réseaux 5G basés sur l'SDN, l'SDN en tant que solution de réseau cloud hybride, les réseaux gérables pour la communication IoT, etc.

2.2.7.1 SDN dans le ≪ Cloud Computing ≫

Le Cloud Computing [41] est une technologie informatique importante, construite autour du concept d'investissement réduit et des solutions de facturation à la carte. Les fournisseurs de services cloud emploient généralement des modèles de ≪ paiement à l'utilisation ≫.
Alors que les méthodes, architectures et techniques de traitement conventionnelles peuvent limiter les performances du centre de données cloud, le Software-Defined Cloud Computing (SDCC) est une approche dans laquelle les services de virtualisation de toutes les ressources réseau d'un Data Center (DC) sont définis par logiciel. Les auteurs de [42] ont expliqué comment Les SDCC résolvent les problèmes des DC traditionnels en fournissant un environnement ouvert permettant aux utilisateurs de gérer les centres de données en fonction de leurs besoins. Ils ont mentionné les avantages qu'une infrastructure SDN peut apporter aux technologies de Cloud Computing, comme la capacité d'accueillir des nouvelles applications, d'améliorer le contrôle de la sécurité et de réduire les coûts de la gestion.

2.2.7.2 SD-WAN

Le réseau étendu (WAN) est l'un des supports de transmission les plus importants pour Internet, c'est un grand réseau informatique qui connecte des groupes d'ordinateurs sur de grandes distances [43]. Des exemples de WAN sont les réseaux inter-centres de données, les réseaux d'entreprise et les réseaux d'opérateurs. Selon les auteurs de [44], Software-Defined Wide Area Network (SD-WAN) est considéré comme la prochaine génération des réseaux WAN, car l'ancien WAN est confronté à des défis dans le monde moderne d'applications émergentes et d'entreprises en croissance. Le SD-WAN utilise la technologie SDN dans l'architecture logique et physique. Dans l'architecture logique, SD-WAN utilise le protocole OpenFlow comme moyen de communication entre la couche de transmission et la couche de contrôle. Dans la couche de transmission, des commutateurs SDN sont installés à la place des commutateurs traditionnels.

2.2.7.3 SDN dans l'IOT

Dans l'IoT (Internet des objets), les objets, dont la nature peut être physique ou virtuelle, ont pour rôle de : détecter, collecter, envoyer, recevoir, communiquer, stocker et traiter des données. Ces objets

sont connectés à Internet pour accomplir des tâches dans différents domaines, tels que : l'agriculture, la santé, l'industrie et même le militaire.
Il a été prouvé que les protocoles ainsi que l'architecture du réseau IoT traditionnel manquent de la capacité, de la mobilité et de l'évolutivité nécessaires pour assurer la fiabilité de la collecte massive des données. Le SDN est considéré comme la nouvelle technologie capable de répondre aux exigences de l'IoT. Les avantages de l'intégration du SDN et de l'IoT ont été reconnus dans plusieurs domaines tels que le transport intelligent et les maisons intelligentes. Pour cela, de nombreuses solutions IoT assistées par SDN ont été proposées sous le nom : « Software Defined Internet of Things » (SDIoT). Ce concept vise à développer et améliorer les architectures IoT assistées par le SDN pour les futures implémentations [45].

2.2.8 Travaux connexes

Dans le contexte du développement et de l'optimisation des solutions SDN, la littérature contient un nombre important des travaux menés sur différents contrôleurs tels que Nox, Pox, Floodlight, OpenDaylight, etc. Notre étude est basée sur l'environnement SDN utilisant un contrôleur RYU, c'est pourquoi nous mentionnons, dans ce qui suit, quelques travaux testés sur ce contrôleur.

2.2.8.1 Travaux de recherche menés sur RYU

Les auteurs dans [46] ont utilisé le contrôleur Ryu pour étudier et évaluer les performances des métriques QoS (Quality of Service) dans le réseau SDN. Ils ont présenté les moyens et les outils pour effectuer leur évaluation. Dans [47], les auteurs ont abordé les fonctionnalités d'évolutivité du contrôleur Ryu en implémentant des scénarios diversifiés dans un environnement expérimental de simulation. Ils ont exposé les étapes permettant de créer un scénario expérimental ainsi que l'analyse des résultats statistiques obtenus, tout en gardant la performance de débit comme objectif principal.
Dans une autre analyse des performances, les auteurs de [48] ont évalué les performances du contrôleur Ryu, après l'étude d'une simulation, à travers les paramètres suivants : la bande passante, le temps aller-retour, la gigue et la perte des paquets.

2.2.8.2 Projets de MASTER réalisés au niveau du centre universitaire de Mila

Dans le projet de Master [49], les étudiants ont développé une application qui assiste les administrateurs dans l'opération de migration des réseaux traditionnels vers les réseaux SDN. La solution « Route-Translator » a été conçue pour minimiser autant que possible l'intervention humaine dans

le processus de migration. L'application a été testée dans un environnement SDN simulé en utilisant mininet [50].

Dans le projet de Master [51], les étudiants ont développé l'application « VNET- Manager » qui simplifie la gestion des réseaux virtuels installés dans une infrastructure SDN. Cette application avait pour but de minimiser l'intervention humaine et, en même temps, d'assurer un meilleur contrôle des différentes opérations liées à la gestion des réseaux virtuels telles que : la création, la mise à jour et la suppression.

2.3 Contrôle du trafic IPv6 : Cas des réseaux SDN utilisant le contrôleur RYU

L'un des avantages de l'architecture SDN est qu'elle élimine les boîtiers de médiation du réseau en les remplaçant par des applications qui s'exécutent au niveau du contrôleur. Le dispositif de pare-feu (firewall) est l'un de ces boîtiers de médiation (voir la **Figure 2.4**). Etant une application SDN, le pare-feu est conçu pour empêcher les paquets entrants non autorisés, les paquets provenant de diverses sources ainsi que les paquets sortants, et ce, en se basant sur les politiques définies par l'administrateur.

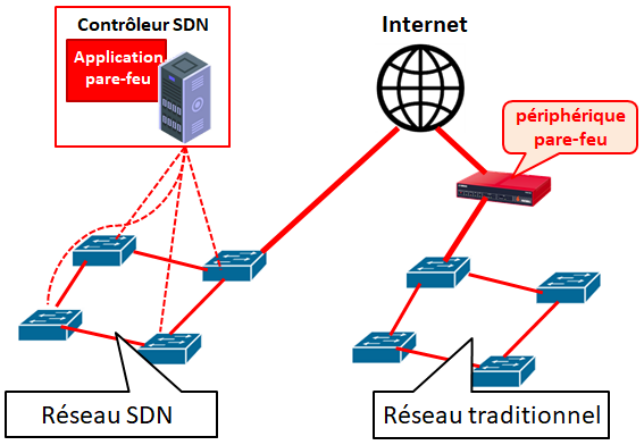

FIGURE 2.4 – Le pare-feu dans les deux réseaux traditionnel et SDN .

Les développeurs de RYU ont renforcé le contrôleur par des applications built-in, dans le but de permettre aux administrateurs d'exploiter ses services. Parmi ces applications, on trouve l'application « rest_firewall » qui représente le noyau de l'étude mené dans ce projet pour le contrôle du trafic IPv6. Afin de pouvoir étudier les différentes fonctionnalités de l'application « rest_firewall », on propose une topologie simplifiée simulée utilisant l'outil mininet (voir la **Figure 2.5**). Cette topologie est composée d'un contrôleur, un seul switch SDN et de trois hôtes.

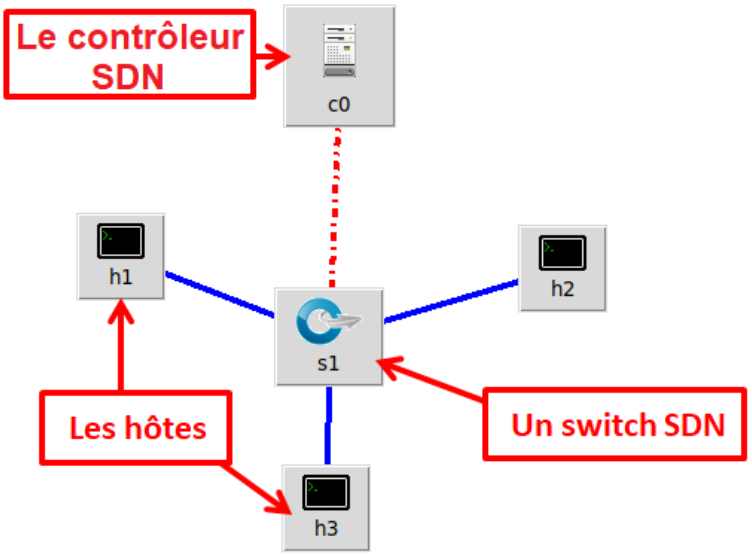

FIGURE 2.5 – Topologie simulée utilisant mininet.

Toutes les machines doivent être configurées avec une adresse IPv6 appartenant à la même adresse réseau. La **Figure 2.6** montre comment attribuer l'adresse « fd00 : : 1/64 » à h1 (Nous avons choisi des adresses de type Unique-Local pour nos simulations) :

```
mininet> h1 ifconfig h1-eth0 inet6 add fd00::1/64
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fd00::1  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::40fd:d0ff:fe65:9751  prefixlen 64  scopeid 0x20<link>
        ether 42:fd:d0:65:97:51  txqueuelen 1000  (Ethernet)
        RX packets 27  bytes 4148 (4.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1172 (1.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 2.6 – Attribution d'une adresse IPv6 à h1.

2.3.1 Simulation du paramétrage de l'application « rest_firewall »

Initialement, l'administrateur lance l'application « rest_firewall », comme le montre la **Figure 2.7** :

```
aymen@aymen-Inspiron-3537:~$ ryu-manager ryu.app.rest_firewall
loading app ryu.app.rest_firewall
loading app ryu.controller.ofp_handler
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app ryu.app.rest_firewall of RestFirewallAPI
instantiating app ryu.controller.ofp_handler of OFPHandler
(4695) wsgi starting up on http://0.0.0.0:8080
[FW][INFO] dpid=0000000000000001: Join as firewall.
```

FIGURE 2.7 – Lancement de rest_firewall.

L'architecture SDN offre à l'administrateur réseau la possibilité d'utiliser des API pour paramétrer ou bien configurer les applications SDN. Diverses plateformes peuvent être utilisées pour communiquer via des API telles que Testfully, Insomnia, ...ect. Postman (voir la **Figure 2.8**) représente la plateforme la plus répandue [52], c'est pourquoi nous l'avons utilisé dans nos simulations sur l'application « rest_firewall ».

FIGURE 2.8 – Une vue partielle de l'interface de postman.

Avant de pouvoir gérer les politiques, dans le but de contrôler le trafic IPv6, il faut d'abord activer le service pare-feu au niveau du switch s1 (voir la **Figure 2.9**).

Les détails concernant le format de l'url :

- 192.168.105.70 : l'adresse ip de contrôleur.
- 8080 : numéro de port.
- 0000000000000001 : id de switch s1.

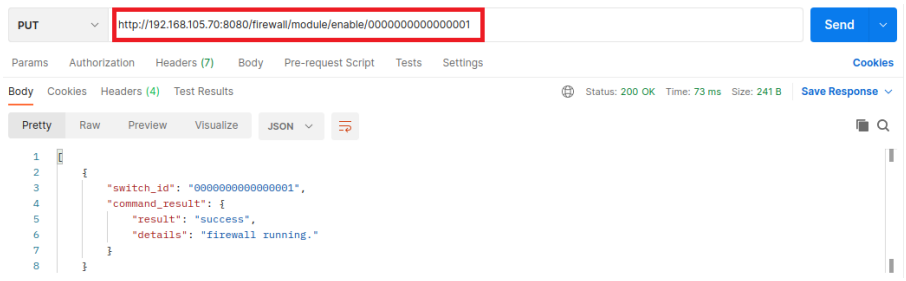

FIGURE 2.9 – Activation du service pare-feu.

A partir de là, l'administrateur est en mesure d'effectuer les opérations suivantes :

• **Ajouter des politiques :**

Pour ajouter une politique, l'administrateur doit formuler un message de type JSON (JavaScript Object Notation) contenant un ensemble de paramètres (un ou plusieurs) qui décrivent le flux à autoriser ou bien à interdire. Ces paramètres incluent : l'adresse IPv6 source, l'adresse IPv6 destination, le protocole utilisé, le numéro de port source, le numéro de port destination, etc

Prenons l'exemple du PING, avant d'ajouter des politiques, le test PING entre h1 et h2 échoue, comme le montre la **Figure 2.10**.

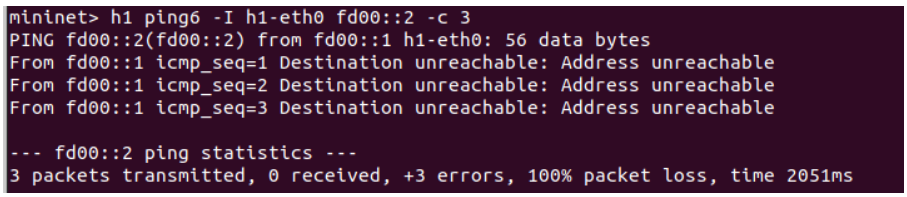

FIGURE 2.10 – Echec du test PING avant l'ajout des politiques.

Pour pouvoir effectuer des tests de communication PING entre les deux machines h1 & h2, l'administrateur doit introduire les quatre politiques suivantes :

— Une politique autorisant le trafic ICMPv6 (Internet Control Message Protocol vesion 6) à partir de la machine h1 vers h2.
— Une politique autorisant le trafic ICMPv6 à partir de la machine h2 vers h1.
— Une politique autorisant le trafic de découverte des voisins à partir de la machine h1 vers h2.
— Une politique autorisant le trafic de découverte des voisins à partir de la machine h1 vers h2.

Un exemple d'ajout d'une politique autorisant le trafic ICMPv6 à partir de la machine h1 vers h2 est détaillé dans la **Figure 2.11**.

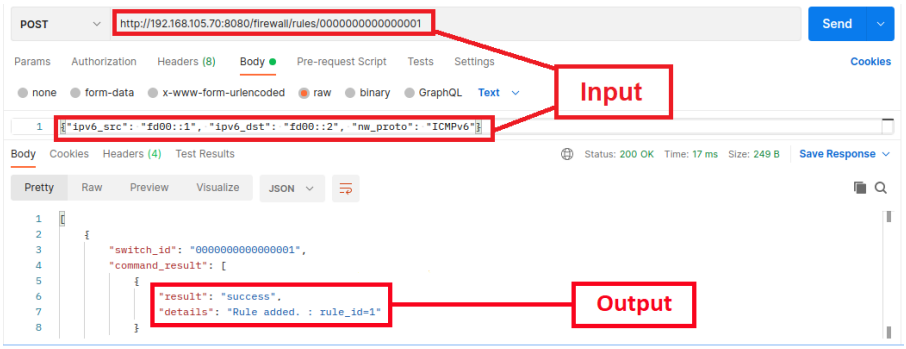

FIGURE 2.11 – Exemple d'ajout d'une politique.

Après avoir introduit les quatre politiques, le test PING entre h1 et h2 sera établi, comme illustré dans la **Figure 2.12** :

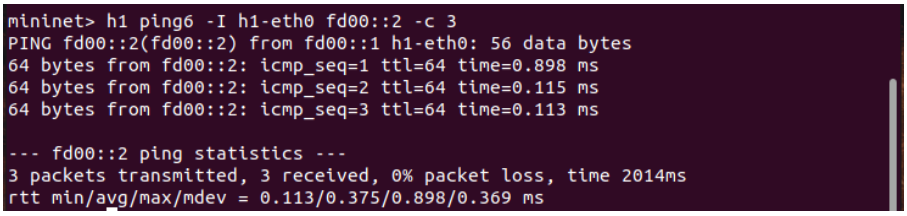

FIGURE 2.12 – Succès du test PING après l'ajout des politiques.

L'application « rest_firewall » permet à l'administrateur de visualiser la liste de toutes les politiques ajoutées ou installées, pour vérification, comme le montre la **Figure 2.13** :

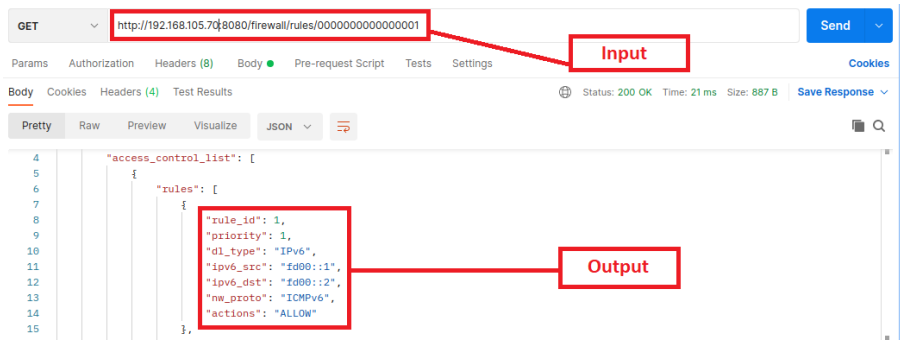

FIGURE 2.13 – Affichage de la liste des politiques ajoutées.

- **Modifier des politiques :**

Les politiques ajoutées sont par défaut autorisées, c'est-à-dire l'action correspondante à ces politiques est « ALLOW». Pour modifier l'action relative à une politique, la même politique doit être introduite

à nouveau avec le paramètre « actions » fixé à la valeur « DENY », comme indiqué dans les **Figure 2.14** :

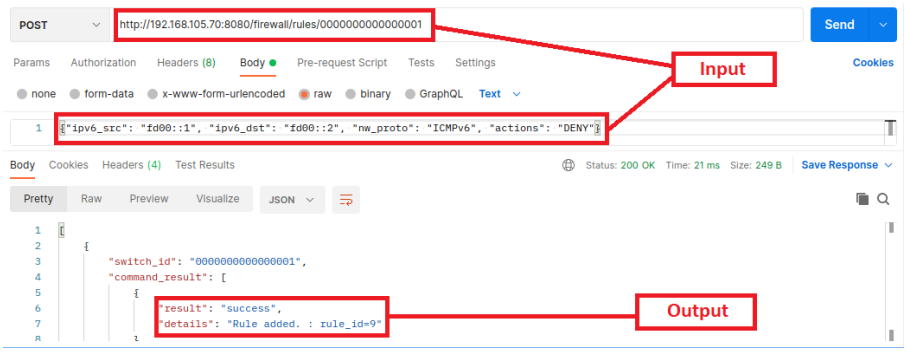

FIGURE 2.14 – Modification de l'action relative à une politique.

L'action relative à la politique est maintenant fixée à la valeur « DENY », comme indiqué dans la **Figure 2.15** :

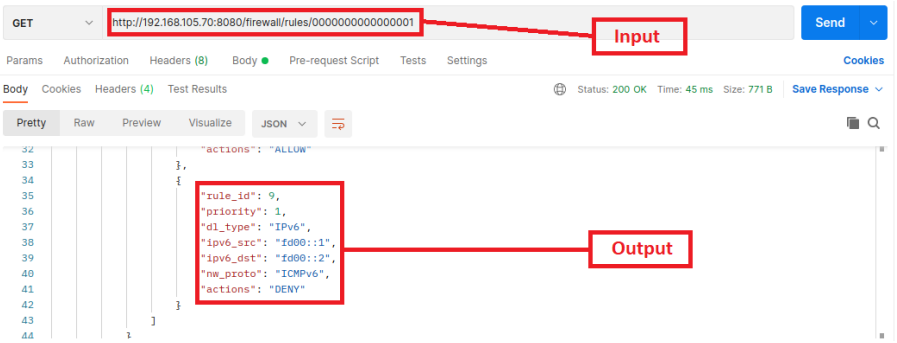

FIGURE 2.15 – Vérification de la politique modifiée.

• **Supprimer des politiques :**

Une politique peut être supprimée en envoyant un message JSON avec la méthode DELETE. Dans le corps de ce message JSON, l'administrateur doit spécifier le paramètre « rule_id » relative à la politique à supprimer, comme indiqué dans la **Figure 2.16** :

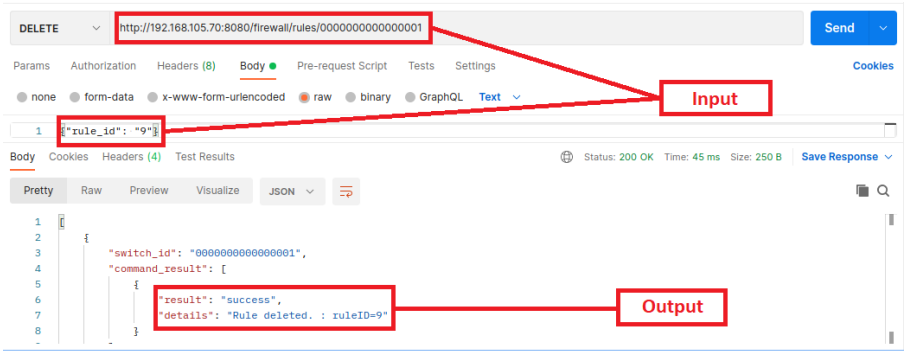

FIGURE 2.16 – Suppression d'une politique.

2.3.2 Problèmes relatives la saisie manuelle

Les simulations présentées précédemment concernent une topologie très simple. Maintenant imaginons un réseau de grande taille composé de milliers de machines et une centaine de switches (par exemple : un réseau d'une entreprise, d'un ministère, d'une université ou autres organismes). Dans ce cas, la configuration ou bien le paramétrage de l'application est sujet à l'erreur. L'administrateur doit faire très attention lors de la saisie des URL ainsi que les messages JSON, car les erreurs peuvent entrainer beaucoup de problème dont la résolution consomme beaucoup de temps, en supposant que l'administrateur est qualifié.

Les erreurs peuvent se produire dans n'importe quelle opération. Dans ce qui suit, nous allons mentionner quelques points importants pour chacun des services de l'application « rest_firewall ».

- **Ajouter des politiques :**

L'envoi d'un message JSON non conforme peut entraîner des problèmes de trafic réseau. Même une petite erreur telle que l'ajout d'un simple espace corrompt la politique. Les **Figure 2.17** et **2.18** montrent le résultat de l'ajout d'une politique corrompue :

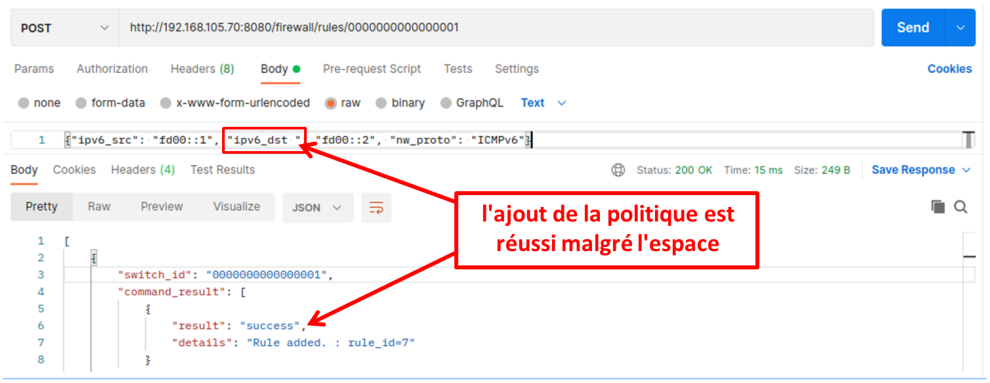

FIGURE 2.17 – Succès de l'ajout d'une politique non conforme.

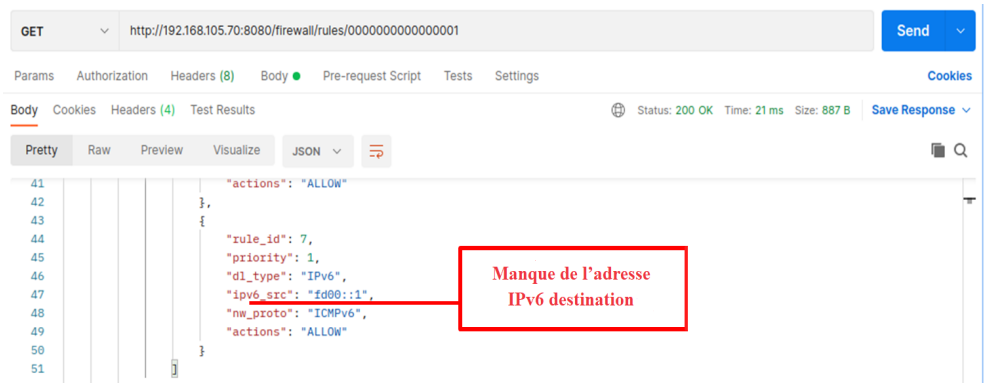

FIGURE 2.18 – L'affichage de la politique confirme le manque d'un paramètre.

- **Modifier des politiques :**

L'administrateur doit connaître les détails exacts de la politique pour pouvoir la modifier. Introduire un message JSON avec un paramètre diffèrent entraîne l'ajout d'une nouvelle politique en plus de celle d'origine, ce qui peut affecter négativement le comportement du réseau.

- **Supprimer des politiques :**

L'administrateur doit connaître le paramètre « rule_id » relatif à une politique pour pouvoir la supprimer. Dans le cas où l'administrateur introduit un « rule_id » erroné, il n'existe aucun mécanisme qui lui permet de restaurer la politique supprimer par erreur, ce qui complique la tâche de mise à jour des politiques.

Pour récapituler, le contrôle manuel du trafic réseau IPv6 est complexe et sujet à l'erreur. Une telle méthode de gestion exige à l'administrateur de créer un journal numérique personnel, afin de pouvoir suivre les différentes opérations réalisées sur le réseau.

2.4 La solution proposée « IPv6 Traffic-Controller »

L'objectif de notre projet est de concevoir une solution qui simplifie aux administrateurs la gestion des politiques relatives au contrôle du trafic IPv6. L'« IPv6 Traffic-Controller » (voir la **Figure 2.19**) représente une solution qui interagit avec l'application « rest_firewall » de manière fiable et bien contrôlée. Elle minimise les interactions manuelles, assure la sauvegarde des différentes opérations dans une base de données et élimine entièrement les erreurs de saisie, car les URLs et les messages JSON sont générés de manière automatique par le système. L'administrateur n'a qu'à spécifier la tâche requise par un simple clic de bouton.

Dans ce qui suit, nous allons exposer les quatre principaux axes fonctionnels de notre solution :

— La planification organisationnelle du réseau.

— La spécification des plages d'adresses IPv6 pour toutes les structures.

— Le paramétrage de l'application « rest_firewall ».

— L'affichage des informations relatives au contrôle du trafic IPv6.

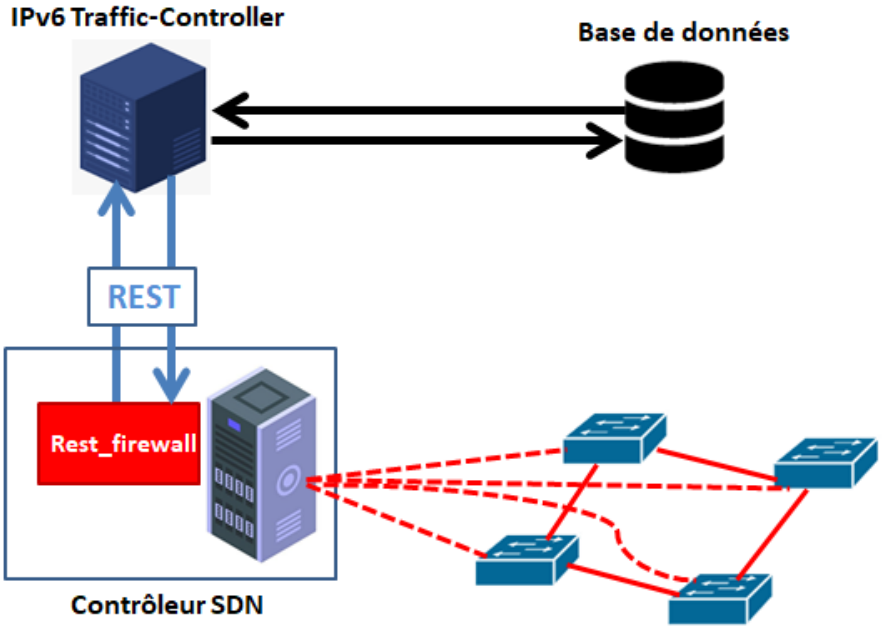

FIGURE 2.19 – Intégration de la solution « IPv6 Trafic-Controller » dans l'architecture SDN.

2.4.1 Planification organisationnelle du réseau

Les entreprises et les sociétés fonctionnent de manière hiérarchique où différentes structures organisationnelles ont différents niveaux de contrôle et de gestion.

L'objectif dans cette phase est de permettre à l'administrateur de définir, au niveau de « IPv6 Traffic-Controller », la structuration de son réseau, reflétant l'organigramme hiérarchique de l'entreprise (ou bien autre organisme) dont laquelle il travaille (voir la **Figure 2.20** & **Figure 2.21**).

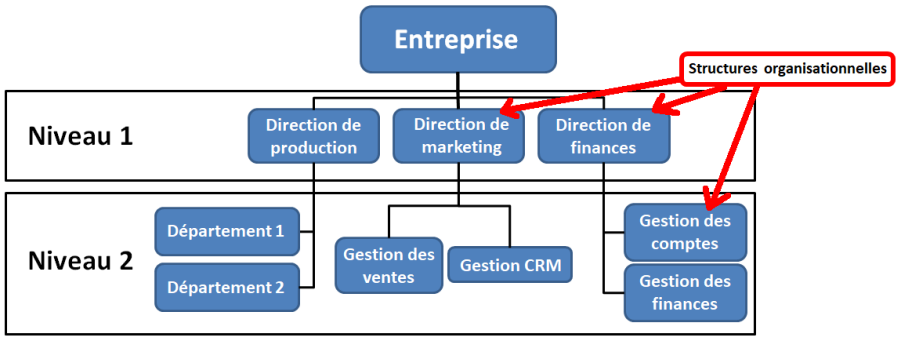

FIGURE 2.20 – Un exemple de la structuration à définir au niveau de « IPv6 Traffic-Controller ».

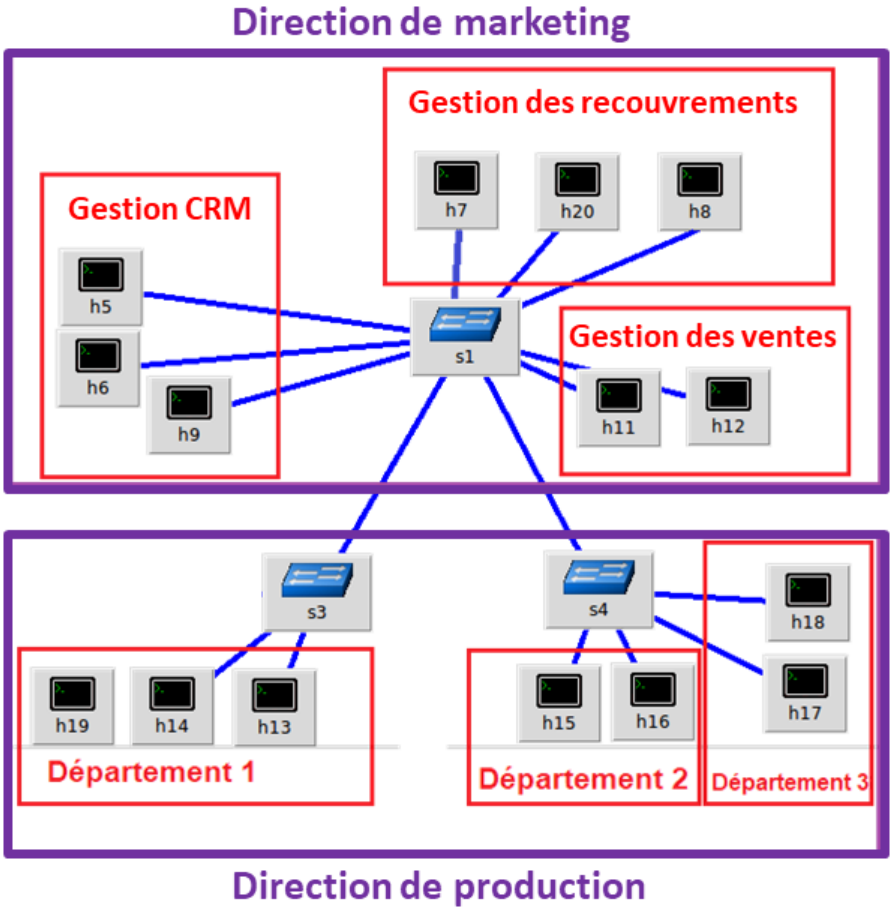

FIGURE 2.21 – Un exemple de la structuration au niveau de la topologie réseau.

Le service « Gestion CRM » et le service « Gestion des ventes » (voir la **Figure 2.21**) appartiennent

hiérarchiquement à la direction de marketing. De la même manière, les autres structures organisationnelles appartiennent à leur structure supérieure. Cette planification permet à l'administrateur de contrôler le trafic entre les structures au lieu de gérer le trafic entre les machines de façon individuelle.

2.4.2 Spécification des plages d'adresses IPv6 pour toutes les structures

C'est très important de noter que l'application « rest_firewall » n'assure pas le routage du trafic réseau. Donc, pratiquement toutes les machines connectées doivent être configurées avec des adresses IPv6 appartenant à la même adresse réseau qu'on a appelé la plage globale.

Pour bien contrôler les communications, chaque structure aura une plage IPv6 spécifique (un sous-ensemble de la plage globale). Ces plages IPv6 doivent respecter la hiérarchie de l'organigramme défini dans la phase de la planification organisationnelle du réseau.

Dans ce qui suit, nous allons détailler le principe de cette phase en simulant la planification d'adressage de l'organigramme précèdent (voir la **Figure 2.20**).

Soit la plage globale : « fd00 :0 :0 : 1 : :/64 ». L'administrateur doit d'abord spécifier, pour chaque niveau, une longueur de préfixe en respectant les points suivant :

- La longueur du préfixe doit être supérieure à 64 et inférieure ou égale à 112.
- La longueur du préfixe doit être hiérarchique pour les structures de différents niveaux.
- La longueur du préfixe de chaque niveau doit être suffisante pour affecter une plage différente à chaque structure de ce niveau.

L'affectation de la longueur de préfixe « 68 » pour le niveau 1 nous donne la possibilité d'avoir au maximum16 structures (directions) différente à ce niveau (voir la **Figure 2.22**). Ceci est calculé comme suit :

$2^{(PNC–PNS)} = 2^{(68–64)} = 2^4 = 16$ directions

PNC : Préfixe du niveau courant.

PNS : Préfixe du 1er niveau supérieur.

Même chose pour le niveau 2. L'affectation de la longueur de préfixe « 72 » nous donne la possibilité d'avoir au maximum 16 structures (département) par direction (voir la **Figure 2.22**), calculé comme suit : $2^{(72–68)} = 2^4 = 16$ département par direction.

Plage global	fd00 : 0000 : 0000 : 0001 : 0000 : 0000 : 0000 : 0000 /64
Niveau 1	fd00 : 0000 : 0000 : 0001 : X000 : 0000 : 0000 : 0000 /68
Niveau 2	fd00 : 0000 : 0000 : 0001 : X Y00 : 0000 : 0000 : 0000 /72

FIGURE 2.22 – Introduction de la longueur du préfixe pour chaque niveau.

Maintenant, l'administrateur affecte une plage d'adresses (préfixe) à chaque structure organisationnelle en respectant ce qui suit :

— La longueur du préfixe désignée pour le niveau de la structure.

— La hiérarchisation des plages d'adresses selon le plan organisationnel (voir les **Figure 2.23** & **2.24**).

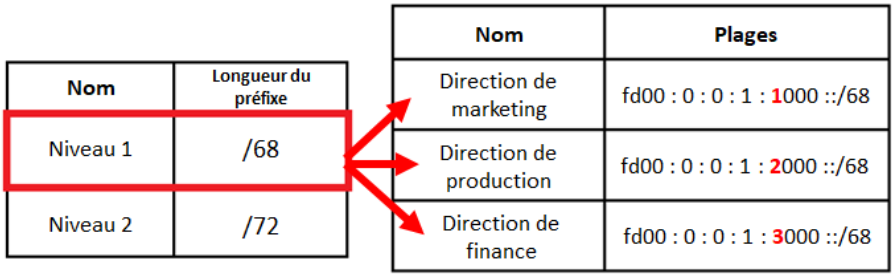

FIGURE 2.23 – Affectation des plages d'adresse pour les structures de niveau 1.

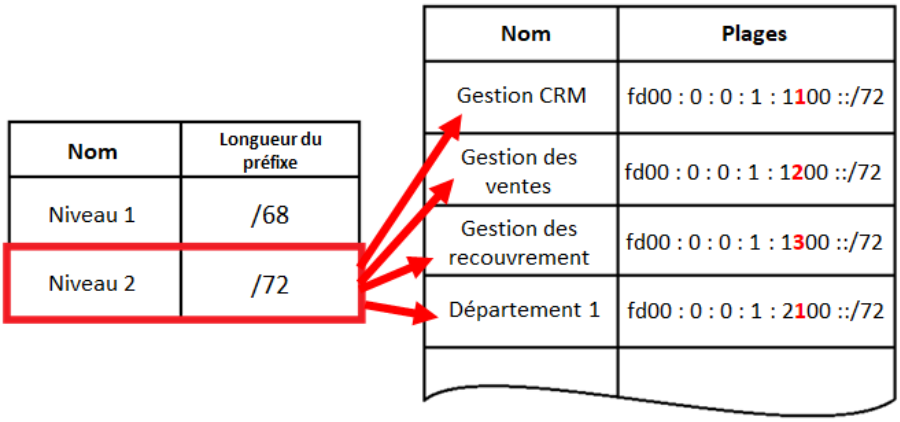

FIGURE 2.24 – Affectation des plages d'adresse pour les structures de niveau 2.

Afin d'assurer la fiabilité du contrôle de trafic, la configuration des adresses IPv6 au niveau des machines du réseau doit respecter les deux règles suivantes :

— Toutes les machines du réseau doivent être configurées avec un masque unifié → « /64 ». C'est la longueur du préfixe de la plage globale.

— Les adresses IPv6 des machines doivent respectées les plages des structures, introduites au niveau de « IPv6 Traffic-Controller » (selon l'appartenance de chaque machine).

L'automatisation de cette opération (la configuration des adresses IPv6 au niveau des machines) n'est pas couverte dans cette étude. Elle fera l'objet d'un futur projet MASTER.

Après la finalisation de cette phase, l'administrateur peut gérer les politiques relatives au contrôle du trafic entre les structures de son réseau à travers « IPv6 Traffic-Controller ».

2.4.3 Le paramétrage de l'application « rest_firewall »

Notre solution « IPv6 Traffic-Controller » assure la communication avec « rest_firewall » à travers les deux informations suivantes :

— L'URL de l'API REST (voir la **Figure 2.25**).

— Le message JSON qui contient les paramètres définissant la politique à ajouter ou bien à supprimer.

URL	IP du contrôleur+ /firewall/rules + l'ID du switch
Méthode	POST | PUT | GET | DELETE

FIGURE 2.25 – Format générique de l'URL de l'API REST relative à « rest_firewall ».

Les URL des API sont générées de manière complètement automatique. La méthode de transmission (POST,DELETE) est, aussi, fixée par le système selon la tache activée (ajout, suppression ou bien affichage). La seule information requise est l'adresse IP du contrôleur SDN. Cette adresse sera introduite manuellement par l'administrateur, une seule fois, au démarrage de « IPv6 traffic-Controller ». Les messages JSON sont générés par le système avec un minimum d'interactions manuelles. Par exemple, dans le cas d'ajout d'une politique, l'administrateur peut spécifier par clic de bouton les paramètres suivants :

— La structure source et la structure destination (voir la **Figure 2.26**) : Cela, simplifie les opérations de contrôle du trafic, car l'administrateur n'aura pas besoin de mémoriser les plages d'adresses IPv6 des différentes structures.

FIGURE 2.26 – Spécification des structures source & destination.

— Le type de trafic à contrôler : Il peut s'agir d'un protocole bien-connu pouvant être spécifié par un clic de bouton, ou bien d'un protocole non connu nécessitant des informations supplémentaires (comme le numéro de port destination (voir la **Figure 2.27**).

FIGURE 2.27 – Spécification du type de trafic à contrôler.

— L'action relative à la politique pouvant être « ALLOW » pour autoriser le trafic ou bien « DENY » pour le refuser. « IPv6 Traffic-Controller » considère l'option « ALLOW » l'action par défaut. Donc, l'administrateur doit déterminer ce paramètre (par clic de bouton) uniquement dans le cas de l'option « DENY ».

2.4.4 L'affichage des informations relatives au contrôle du trafic IPv6

L'objectif de cette fonctionnalité est de fournir à l'administrateur ce qui suit :

— Un service de recherche permettant à l'administrateur d'explorer les informations stockées dans la base de données en introduisant des mots clés dans une barre de recherche (voir la **Figure 2.28**).

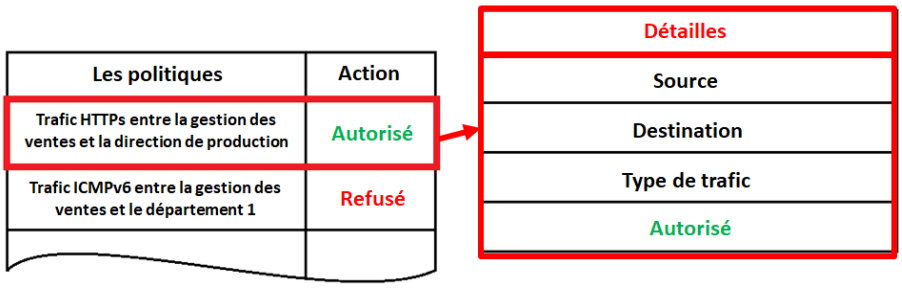

FIGURE 2.28 – L'affichage de la liste complète des politiques.

— Un service de recherche des politiques simplifié à travers la visualisation des structures organisationnelles sous forme d'arborescence. Comme le montre la Figure 2.29, en sélectionnant les

structures source et destination, le système affiche toutes les politiques combinant ces structures.

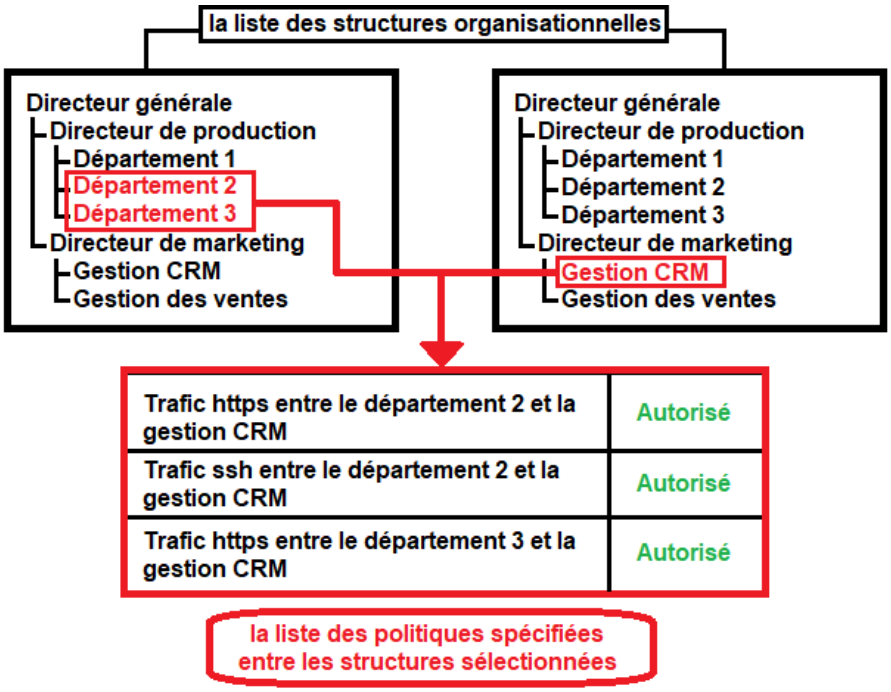

FIGURE 2.29 – Affichage des politiques spécifiées entre des structures sélectionnées.

2.5 Conclusion

Dans ce chapitre, nous avons introduit la technologie SDN. Puis, nous avons exposé la problématique liée à l'interaction manuelle avec les applications SDN permettant de gérer le réseau. Dans la dernière partie de ce chapitre, nous avons présenté la solution « IPv6 Traffic-Controller » qui traite le contrôle du trafic IPv6 dans un environnement SDN utilisant le contrôleur RYU.
Dans le chapitre suivant, nous allons détailler la conception de notre système, ensuite nous allons présenter l'environnement de travail qui nous a permis de réaliser ce projet.

CHAPITRE 3

CONCEPTION ET RÉALISATION

3.1 Introduction

La modélisation d'une application représente une étape cruciale dans le développement de n'importe quel projet logiciel. Elle nous permet de mieux comprendre son fonctionnement et de maitriser la complexité du système à réaliser.
Nous commençant, dans ce chapitre, par la modélisation de notre solution « IPv6 Traffic-Controller » utilisant le langage UML qui s'est imposé comme une norme standard dans la conception orientée objets. Ensuite, nous allons spécifier l'environnement de développement (Langages de programmation, bibliothèques, utilitaires . . .) qui nous a permis de réaliser ce projet.

3.2 Conception de « IPv6 Traffic-Controller »

Le développement d'une application ou d'un logiciel implique généralement le passage par un ensemble d'étapes de modélisation. Actuellement il y'a plusieurs méthodes de conception, parmi lesquelles on trouve l'UML.

3.2.1 Définition d'UML

UML (Unified Modeling Language) est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions, lancé par l'OMG (Object Management Group), dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets [53].

3.2.2 Les vues et les diagrammes UML

UML dans sa 2 éme édition fournit 13 diagrammes, chacun d'eux étant dédié à la représentation d'un concept spécifique Ces types de graphiques sont divisés en trois vues classiques (voir la figure 3.2).

vue fonctionnelle : La vue fonctionnelle vise à appréhender les interactions entre les différents acteurs/utilisateurs et le système, d'une part sous la forme d'objectifs à atteindre et d'autre part sous la forme d'un enchaînement chronologique de scénarios.

FIGURE 3.1 – Logo UML [54].

vue structurelle : Une vue structurelle ou statique s'occupe de la structure des données et tente d'identifier les objets qui composent un programme, leurs propriétés, opérations et méthodes, ainsi que les liens ou associations qui les unissent.

vue dynamique : Cette vue se concentre davantage sur les algorithmes et le "traitement". Elle vise à décrire l'évolution (la dynamique) des objets complexes d'un programme tout au long de leur durée de vie. De leur naissance à leur mort, les objets voient leurs changements d'état guidés par les interactions avec d'autres objets.

3.2.3 Processus unifié

3.2.3.1 Définition d'un processus de développement logiciel

Un processus décrit une série d'étapes partiellement ordonnées qui contribuent au développement d'un système existant ou à l'acquisition d'un système logiciel [56].

3.2.3.2 Définition du processus unifié (UP)

Le processus unifié est une approche itérative de développement logiciel centrée sur l'architecture, guidée par des cas d'utilisation et axée sur la réduction des risques. Il s'agit d'un ensemble de procédures qui peuvent être adaptées à un large éventail de systèmes logiciels , à divers domaines

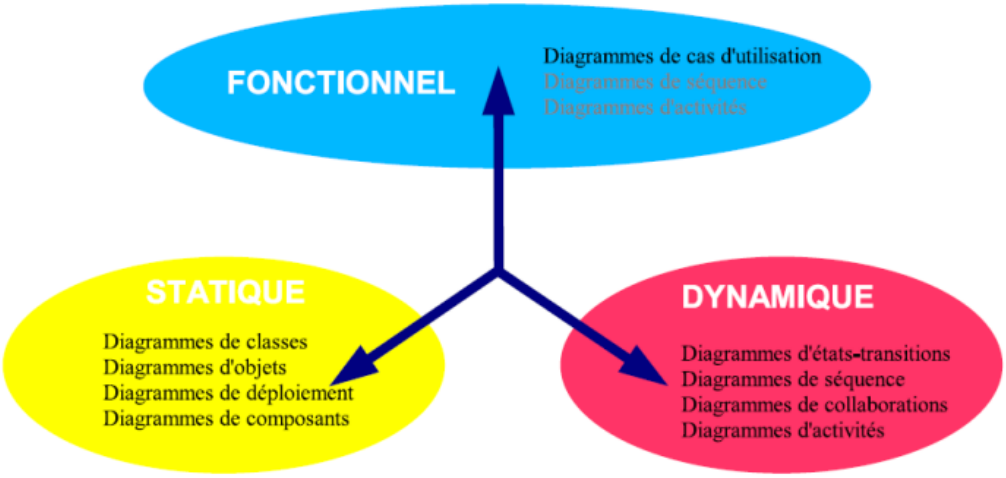

FIGURE 3.2 – Les vues et les diagrammes UML [55].

d'application , à divers modèles commerciaux , à différents niveaux de compétence et à diverses tailles d'entreprise [57].
Dans la modélisation de notre système, nous adopterons une approche basée sur le processus UP , en passant par les trois étapes suivantes :

- **Identification des besoins :**
  - Diagramme de cas d'utilisation
  - Diagramme de séquence système

- **Phase d'analyse :**
  - Diagramme d'activités

- **Phase de conception :**
  - Diagramme de classes

A-Identification des besoins : L'objectif de cette tâche est d' identifier les services et les taches du système étudié :

A.1-Diagramme de cas utilisation

• Définition :

Ce diagramme est destiné à représenter la façon dont les besoins des utilisateurs sont liés au système [58].

• L'objectif :

C'est la première étape de l'analyse UML. Elle permet de :

— Modéliser les besoins des utilisateurs.

— Identifier les fonctions principales ou critiques et les limites du système.

— Représenter les différentes manières dont un utilisateur peut interagir avec le système.

• Les éléments de diagramme de cas utilisation

1.Acteur :

Un acteur représente un rôle joué par une entité externe (comme un utilisateur humain, un dispositif matériel ou un autre système) qui interagit directement avec le système étudié [58]. La représentation graphique standard de l'acteur en UML est l'icon appelé « stick man » avec le nom de l'acteur sous le dessin. Il ya deux type d'acteurs :

Un acteur principal :

— Directement concerné par le cas d'utilisation décrit.

— Sollicite le système pour obtenir un résultat perceptible.

Un acteur secondaire :

— Il est sollicité pour des informations complémentaires.

— Nécessaire au déroulement du cas d'utilisation décrit.

2.Cas d'utilisation :

Représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier [58].

3.Les relations :

— **Relation d'inclusion :** Une relation d'inclusion d'un cas d'utilisation 'A' par rapport à un cas d'utilisation 'B' signifie que le comportement décrit par le cas 'A' inclut le comportement du cas 'B'. Elle est représentée par une flèche discontinu « inclusion ».

— **Relation d'extension :** Une relation d'extension d'un cas d'utilisation 'A' par un cas d'utilisation 'B' signifie que le cas d'utilisation 'A' peut être appelé au cours de l'exécution du cas d'utilisation 'B'.

— **Relation de généralisation :** Un cas 'A' est une généralisation du cas 'B' si 'B' est un cas particulier de 'A'. Cette relation de généralisation/spécialisation existe dans la plupart des diagrammes UML et conduit au concept d'héritage dans les langages orientés objet.

Dans notre système, l'acteur principal est le déclencheur de tous les cas d'utilisation et qui interagit avec l'application c'est l'administrateur réseau.

L'acteur secondaire est le contrôleur SDN qui est nécessaire pour le déroulement des cas utilisations.

Les services offerts par « **IPv6 Traffic-Controller** » (comme le montre la **Figure 3.3**) sont résumés dans les cas d'utilisations suivants :

— S'authentifier.

— Introduire l'adresse IP du contrôleur.

— Introduire le préfixe global du réseau IPv6.

— Gérer les niveaux hiérarchiques.

— Gérer les structures organisationnelles.

— Gérer les politiques réseau.

— MAJ les niveaux hiérarchiques.

— MAJ les structures organisationnelles.

— Chercher des politiques.

— Réinitialiser les politiques.

— Réinitialiser le réseau.

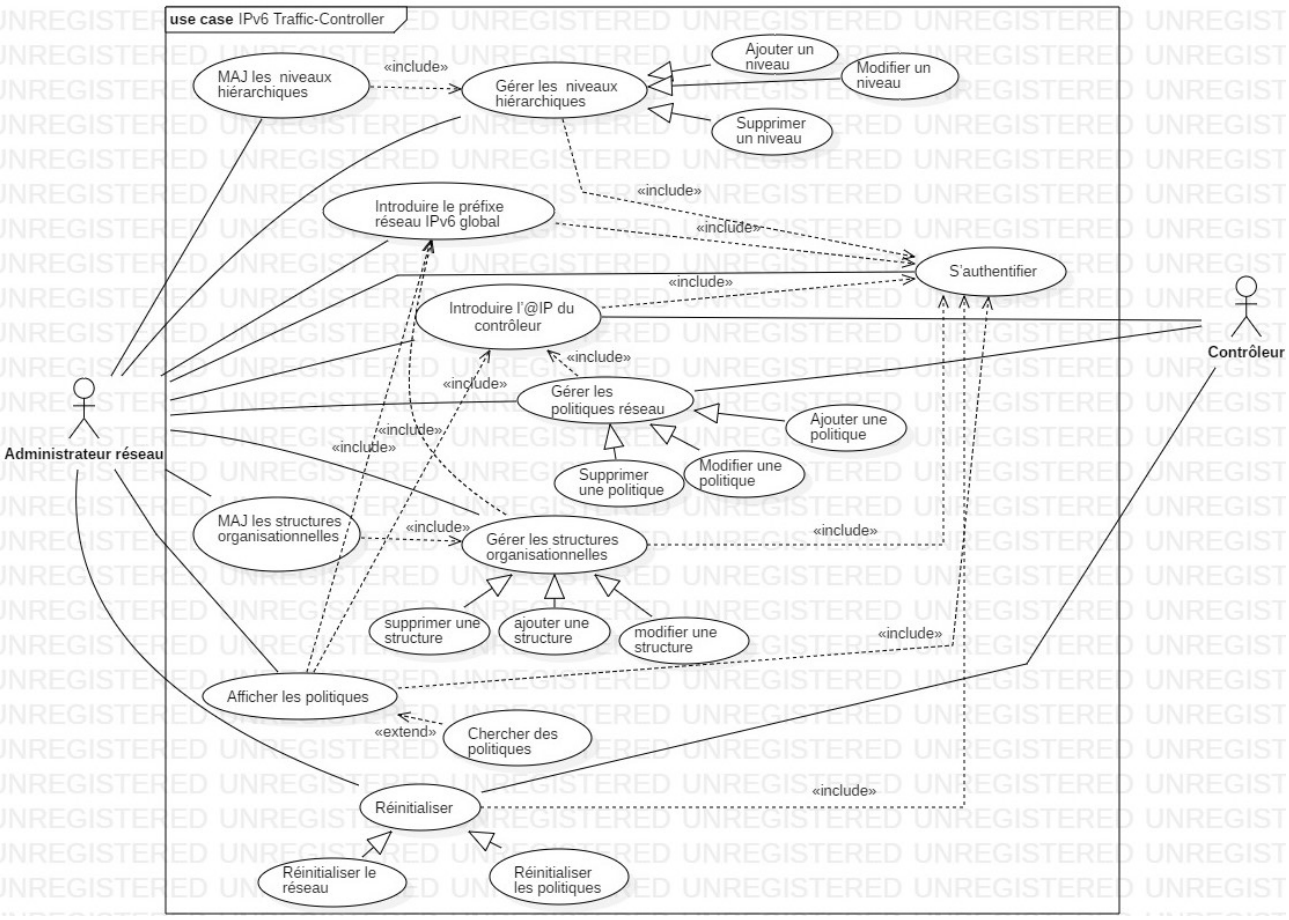

FIGURE 3.3 – Diagramme de cas d'utilisation du « IPv6 Traffic-Controller ».

Description textuelle d'un cas utilisation :

Chaque cas d'utilisation doit être associé à une description textuelle des interactions entre l'acteur, le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs.

La description textuelle d'un cas d'utilisation est organisée en six points :

- **Objectif :** Décrivez brièvement le contexte et les résultats attendus du cas d'utilisation .
- **Acteurs concernés :** Le ou les acteurs concernés par le cas doivent être identifiés en précisant globalement leur rôle (acteur primaire et/ou secondaire).
- **Pré-conditions :** Si certaines conditions particulières sont requises avant l'exécution du cas, elles sont à exprimer à ce niveau.
- **Post-condition :** Par symétrie, si certaines conditions particulières doivent être réunies après l'exécution du cas, elles sont à exprimer à ce niveau.

— **Scénario nominal :**Il s'agit là du scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité.

— **Scénarios alternatifs :** Les autres scénarios, secondaires ou correspondant à la résolution d'anomalies, sont à décrire à ce niveau. Le lien avec le scénario principal se fait à l'aide d'une numérotation hiérarchisée (1.1a, 1.1b...) rappelant le numéro de l'action concerné [58].

• Cas d'utilisation ≪ S'authentifier ≫ :

Cas d'utilisation	S'authentifier.
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur réseau d'accéder à ≪IPv6 Traffic-Controller≫.
Pré condition	Le système est connecté à la base de données.
Scénario nominal	1- Le système affiche les champs de saisie relatifs au nom d'utilisateur et le mot de passe de l'administrateur réseau. 2- L'administrateur introduit le nom d'utilisateur et le mot de passe. 3- L'administrateur valide la saisie. 4- Le système vérifie la validité du nom d'utilisateur et le mot de passe. 5- Le système affiche la fenêtre principal.
Alternatives	4.A- Une des informations introduites (nom d'utilisateur/mot de passe) sont erronés ou la connexion du système à la base de données a échoué. Le système affiche un message d'erreur et retourne au scénario nominal à l'étape 2.

TABLE 3.1 – Scénario du cas d'utilisation ≪ S'authentifier ≫.

• **Cas d'utilisation ≪ Introduire l'adresse IP du contrôleur ≫ :**

Cas d'utilisation	Introduire l'adresse IP du contrôleur.
Acteur(s)	L'administrateur réseau.
Objectif	Tester la connectivité du contrôleur SDN.
Pré condition	-Le contrôleur SDN est joignable à partir du réseau.. -L'administrateur réseax est déjà authentifié.
Scénario nominal	1- L'administrateur clique sur le bouton "Connecter "dans l'interface principale. 2- L'administrateur introduit l'adresse IP du contrôleur SDN. 3- L'administrateur valide la saisie. 4- Le système vérifie la validité de l'adresse IP, ensuite confirme la connectivité de l'adresse IP introduite.
Alternatives	4.A-L'adresse IP introduite est erronée ou la connexion du système au contrôleur a échoué. Le système affiche un message d'erreur et retourne au scénario nominal à l'étape 2.

TABLE 3.2 – Scénario du cas d'utilisation ≪ Introduire adresse IP du contrôleur ≫.

• **Cas d'utilisation ≪ Introduire le préfixe global du réseau IPv6 ≫ :**

Cas d'utilisation	Introduire le préfixe global du réseau IPv6.
Acteur(s)	L'administrateur réseau.
Objectif	Déterminer le préfixe de l'adresse IPV6 global du réseau SDN.
Pré condition	-L'administrateur réseaux est déjà authentifié au niveau de l'application.
Scénario nominal	1- L'administrateur introduit le préfixe IPV6 du réseau SDN dans l'interface principale. 2- L'administrateur valide la saisie. 3- Le système vérifie la validité du préfixe IPV6. 4- Le système sauvegarde le préfixe IPv6 dans la base de donnée.
Alternatives	3.A-Le préfixe réseau introduit est erroné. Le système retourne au scénario nominal à l'étape 1.

TABLE 3.3 – Scénario du cas d'utilisation ≪ Introduire le préfixe global du réseau IPv6 ≫.

• **Cas d'utilisation ≪ Ajouter un niveau ≫ :**

Cas d'utilisation	Ajouter un niveau.
Acteur(s)	L'administrateur réseau.
Objectif	Définir les niveaux hiérarchique de l'organisation.
Pré condition	-L'administrateur réseaux est déjà authentifié.
Scénario nominal	1- L'administrateur clique sur le bouton "Niveaux" dans la barre latérale. 2- Le système affiche l'interface du gestion des niveaux organisationnel. 3- L'administrateur clique sur le bouton "Ajouter". 4- Le système affiche une fenêtre de dialogue. 5- L'administrateur saisir les informations du nouveau niveau. 6- L'administrateur valide la saisie. 7- Le système vérifie la validité des informations saisies puis il le ajoute au niveau de l'application.
Alternatives	7.A- Un des champs saisis comporte une information invalide. Le système affiche un message d'erreur et retourne au scénario nominal à l'étape 5.

TABLE 3.4 – Scénario du cas d'utilisation ≪ Ajouter un niveau ≫.

• Cas d'utilisation ≪ Modifier un niveau ≫ :

Cas d'utilisation	Modifier un niveau.
Acteur(s)	L'administrateur réseau.
Objectif	Modifier les niveaux hiérarchique existants.
Pré condition	-La liste des niveaux dans l'application n'est pas vide.
Scénario nominal	1- L'administrateur sélectionne un niveau a partir de l'interface de la gestion des niveaux puis clique sur le bouton "Modifier". 2- Le système affiche une fenêtre contenant les informations du niveau sélectionné. 3- L'administrateur effectue des modifications. 4- L'administrateur valide la saisie. 5-Le système vérifie la validité des informations saisies puis il appliques au niveau de l'application.
Alternatives	5.A- Un des champs saisis comporte une information invalide. Le système affiche un message d'erreur et retourne au scénario nominal l'étape 3.

TABLE 3.5 – Scénario du cas d'utilisation ≪ Modifier un niveau ≫.

• Cas d'utilisation ≪ Supprimer un niveau ≫ :

Cas d'utilisation	Supprimer un niveau .
Acteur(s)	L'administrateur réseau.
Objectif	Supprimer un niveau existante.
Pré condition	-La liste des niveaux dans l'application n'est pas vide.
Scénario nominal	1-L'administrateur clique sur le bouton "Supprimer". 2- Le système supprime le dernier niveau.

TABLE 3.6 – Scénario du cas d'utilisation ≪ Supprimer un niveau ≫.

• **Cas d'utilisation ≪ MAJ les niveaux hiérarchiques ≫ :**

Cas d'utilisation	MAJ les niveaux hiérarchiques .
Acteur(s)	L'administrateur réseau.
Objectif	Sauvegarder les MAJ effectuées (ajout, modification ou suppression) dans la base de données.
Pré condition	- La liste des niveaux dans l'application n'est pas vide.
Scénario nominal	1-L'administrateur clique sur le bouton "mettre à jour". 2- Le système sauvegarde les MAJ effectuées (ajout, modification ou suppression) dans la base de données.
Post condition	L'administrateur ne peut plus modifier la liste des niveaux.

TABLE 3.7 – Scénario du cas d'utilisation ≪ MAJ les niveaux hiérarchiques ≫.

• **Cas d'utilisation ≪ Ajouter une structure ≫ :**

Cas d'utilisation	Ajouter une structure .
Acteur(s)	L'administrateur réseau.
Objectif	Définir les structures organisationnelles dans les niveaux hiérarchiques.
Pré condition	-L'administrateur réseau est déjà authentifié. - Le préfixe globale est déjà définie. - Les niveaux sont déjà sauvegarder dans la base de donnée.
Scénario nominal	1- L'administrateur clique sur le bouton "Structure" dans la barre latérale. 2- Le système affiche l'interface de la gestion des structures organisationnelles. 3- L'administrateur clique sur le bouton "Ajouter". 4- Le système affiche une fenêtre de dialogue. 5- L'administrateur saisir les informations de la nouvelle structure. 6- L'administrateur valide la saisie. 7- Le système vérifie la validité des informations saisies puis il le ajoute au niveau de l'application..
Alternatives	7.A- Un des champs saisis comporte une information invalide. Le système affiche un message d'erreur et retourne au scénario nominal à l'étape 5.

TABLE 3.8 – Scénario du cas d'utilisation ≪ Ajouter une structure ≫.

• **Cas d'utilisation ≪ Modifier une structure ≫ :**

Cas d'utilisation	Modifier une structure .
Acteur(s)	L'administrateur réseau.
Objectif	modifié une structure existante.
Pré condition	-L'administrateur réseau est déjà authentifié. - La liste des structures dans l'application n'est pas vide.
Scénario nominal	1-L'administrateur sélectionne une structure puis clique sur le bouton "Modifier". 2- Le système affiche une fenêtre contenant les informations de la structure. 3- L'administrateur effectue des modifications 4- L'administrateur valide la saisie. 5- Le système vérifie la validité des informations saisies puis il les sauvegardes.
Alternatives	5.A-Un des champs saisis comporte une information invalide. Le système affiche un message d'erreur et retourne au scénario nominal à l'étape 3.

TABLE 3.9 – Scénario du cas d'utilisation ≪ Modifier une structure ≫.

• **Cas d'utilisation ≪ Supprimer une structure ≫ :**

Cas d'utilisation	Supprimer une structure .
Acteur(s)	L'administrateur réseau.
Objectif	Supprimer une structure existante.
Pré condition	- L'administrateur réseau est déjà authentifié. - La liste des structures dans l'application n'est pas vide.
Scénario nominal	1-L'administrateur sélectionne une structure puis clique sur le bouton "Supprimer". 2- Le système supprime la structure sélectionnée ainsi que toutes ses branches inférieures (sous-structures).

TABLE 3.10 – Scénario du cas d'utilisation ≪ Supprimer une structure ≫.

• **Cas d'utilisation ≪ MAJ les structures organisationnelles ≫ :**

Cas d'utilisation	MAJ les structures organisationnelles .
Acteur(s)	L'administrateur réseau.
Objectif	Sauvegarder les MAJ effectuées (ajout, modification ou suppression) dans la base de données.
Pré condition	-L'administrateur réseau est déjà authentifié. -La liste des structures dans l'application n'est pas vide.
Scénario nominal	1-L'administrateur clique sur le bouton "mettre à jour". 2- Le système sauvegarde les MAJ effectuées (ajout, modification ou suppression) dans la base de données.
Post condition	L'administrateur ne peut plus modifier la liste des structures.

TABLE 3.11 – Scénario du cas d'utilisation ≪ MAJ les structures organisationnelles ≫.

• **Cas d'utilisation ≪ Ajouter une politique ≫ :**

Cas d'utilisation	Ajouter une politique .
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur d'ajouter des politiques.
Pré condition	-L'administrateur réseaux est déjà authentifié. -Le système est connecté à la base de données. -Le contrôleur SDN est joignable à partir du réseau. -Les niveaux sont déjà définis. -Les structures sont déjà définis.
Scénario nominal	1- L'administrateur clique sur l'option "Trafic" dans la barre latérale. 2- Le système affiche une interface permettant à l'administrateur de sélectionner des structures entre lesquels des politiques seront ajoutées. 3- L'administrateur sélectionne des structures. 4- L'administrateur clique sur le bouton " Suivant ". 5- Le système affiche l'interface de la gestion des politiques. 6- L'administrateur clique sur le bouton "Ajouter". 7- Le système affiche une fenêtre de dialogue. 8- L'administrateur saisit les informations de la nouvelle politique. 9- L'administrateur valide la saisie. 10- Le système vérifie la validité des informations saisies puis il l'ajoute. 11- Le système transmet la commande relative à la politique au contrôleur SDN et enregistre les détails de cette politique dans la base de données.
Alternatives	10.A- Un des champs saisis comporte une information invalide ou problème de connexion avec le contrôleur. Le système affiche un message d'erreur et retourne au scénario nominal à l'étape 8.

TABLE 3.12 – Scénario du cas d'utilisation ≪ Ajouter une politique ≫.

• **Cas d'utilisation ≪ Afficher les politiques ≫ :**

Cas d'utilisation	Afficher les politiques.
Acteur(s)	L'administrateur réseau.
Objectif	Permet à l'administrateur d'afficher la liste des politiques.
Pré condition	-L'administrateur réseaux est déjà authentifié. -Le système est connecté à la base de données. -Le contrôleur SDN est joignable à partir du réseau.
Scénario nominal	1- L'administrateur clique sur l'option "Trafic" dans la barre latérale. 2- Le système affiche une interface permettant à l'administrateur de sélectionner les structures concernées. 3- L'administrateur sélectionne des structures. 4- L'administrateur clique sur le bouton " Suivant ". 5- Le système affiche l'interface de la gestion des politiques. 6- L'administrateur clique sur le bouton "Afficher". 7- Le système affiche les politiques reliant les structures sélectionnées.

TABLE 3.13 – Scénario du cas d'utilisation ≪ Afficher les politiques ≫.

• **Cas d'utilisation ≪ Modifier une politique ≫ :**

Cas d'utilisation	Modifier une politique .
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur de modifier des politiques.
Pré condition	-Le contrôleur SDN est joignable à partir du réseau. -Le système est connecté à la base de données. -L'administrateur réseau est déjà authentifié. -La liste des politiques dans la base de données n'est pas vide.
Scénario nominal	1- L'administrateur clique sur l'option "Trafic" dans la barre latérale. 2- Le système affiche une interface permettant à l'administrateur de sélectionner les structures concernées. 3- L'administrateur sélectionne des structures. 4- L'administrateur clique sur le bouton " Suivant ". 5- Le système affiche l'interface de la gestion des politiques. 6- L'administrateur clique sur le bouton "Afficher". 7- Le système affiche les politiques. 8- L'administrateur sélectionne une politique. 9- L'administrateur clique sur le bouton "Modifier". 10- Le système affiche une fenêtre de dialogue. 11- L'administrateur effectue des modifications sur la politique. 12- L'administrateur valide le changement. 13- Le système transmet la commande relative à la politique modifiée au contrôleur SDN et enregistre les détails de cette politique dans la base de données.

TABLE 3.14 – Scénario du cas d'utilisation ≪ Modifier une politique ≫.

• **Cas d'utilisation ≪ Supprimer une politique ≫ :**

Cas d'utilisation	Supprimer une politique .
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur de supprimer des politiques.
Pré condition	-Le contrôleur SDN est joignable à partir du réseau. -Le système est connecté à la base de données. -L'administrateur réseau est déjà authentifié. - La liste des politiques dans la base de données n'est vide.
Scénario nominal	1- L'administrateur clique sur l'option "Trafic" dans la barre latérale. 2- Le système affiche une interface permettant à l'administrateur de sélectionner les structures concernées. 3- L'administrateur sélectionne des structures. 4- L'administrateur clique sur le bouton "Suivant ". 5- Le système affiche l'interface de la gestion des politiques. 6- L'administrateur clique sur le bouton "Afficher". 7- Le système affiche les politiques. 8- L'administrateur sélectionne une politique. 9- L'administrateur clique sur le bouton "Supprimer". 10- Le système transmet une commande de suppression relative à la politique au contrôleur SDN et supprime cette politique dans la base de données.

TABLE 3.15 – Scénario du cas d'utilisation ≪ Supprimer une politique ≫.

• **Cas d'utilisation ≪ Réinitialiser les politiques ≫ :**

Cas d'utilisation	Réinitialiser les politiques.
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur réseau de supprimer tout les politiques.
Pré condition	-Le contrôleur SDN est joignable à partir du réseau. -L'administrateur réseau est déjà authentifié.
Scénario nominal	1- L'administrateur clique sur le bouton " Réinitialiser " dans la barre latérale de l'application. 2- Le système affiche une fenêtre de dialogue. 3- L'administrateur sélectionne l'option "Réinitialiser les politiques" dans la fenêtre. 4- L'administrateur clique sur le bouton "Confirmer". 5- Le système supprime toutes les politiques du contrôleur SDN. 6- Le système supprime toutes les politiques de la base de données.

TABLE 3.16 – Scénario du cas d'utilisation ≪ Réinitialiser les politiques ≫.

• **Cas d'utilisation ≪ Réinitialiser le réseau ≫ :**

Cas d'utilisation	Réinitialiser le réseau.
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur réseau de supprimer toutes les politiques, les structures, les niveaux et l'adresse ip global.
Pré condition	-Le contrôleur SDN est joignable à partir du réseau. -L'administrateur réseau est déjà authentifié.
Scénario nominal	1- L'administrateur clique sur le bouton " Réinitialiser " dans la barre latérale de l'application. 2- Le système affiche une fenêtre. 3- L'administrateur sélectionne l'option "Réinitialiser le réseau " dans la fenêtre. 4- L'administrateur clique sur le bouton "Confirmer". 5- Le système supprime toutes les politiques du contrôleur SDN. 6- Le système supprime toutes les politiques, les structures, les niveaux et le préfixe global de la base de données.

TABLE 3.17 – Scénario du cas d'utilisation ≪ Réinitialiser le réseau ≫.

• **Cas d'utilisation ≪ Chercher des politique ≫ :**

Cas d'utilisation	Chercher des politique.
Acteur(s)	L'administrateur réseau.
Objectif	Permet à L'administrateur réseau de rechercher des politiques.
Pré condition	-Le contrôleur SDN est joignable à partir du réseau. -L'administrateur réseau est déjà authentifié.
Scénario nominal	1- L'administrateur clique sur le bouton " Trafic " dans la barre latérale de l'application. 2- Le système affiche une interface permettant à l'administrateur de sélectionner des structures concernée. 3- L'administrateur sélectionner des structures. 4- L'administrateur clique sur le bouton "Suivant". 5- Le système affiche l'interface de la gestion des politiques. 6- L'administrateur choisir le paramètre par lequel effectuer la recherche. 7- L'administrateur saisir dans la barre de recherche. 8- L'administrateur clique sur le bouton "Rechercher". 9- Le système affiche la résultat de la recherche.

TABLE 3.18 – Scénario du cas d'utilisation ≪ Chercher des politique ≫.

A.2-Diagramme de séquence système :

• Définition :

Le diagramme de séquence permet de décrire les scénarios de chaque cas d'utilisation en représentant temporellement les interactions entre les objets ainsi que les messages échangés entre les objets et les acteurs.

• Les composants d'un diagramme de séquence :

— **Scénario :** Représente une série spécifique d'enchaînements qui s'exécutent du début à la fin du cas d'utilisation. Chaque enchaînement permet de décrire les séquences d'actions.

— **Les linges de vie :** Représente l'ensemble des opérations exécutées par un objet.

— **Message :** Permet de modéliser la circulation des informations entre objets, ou bien entre un acteur et un objet. Il est représenté une flèche horizontale.

• Cas d'utilisation ≪ S'authentifier ≫ :

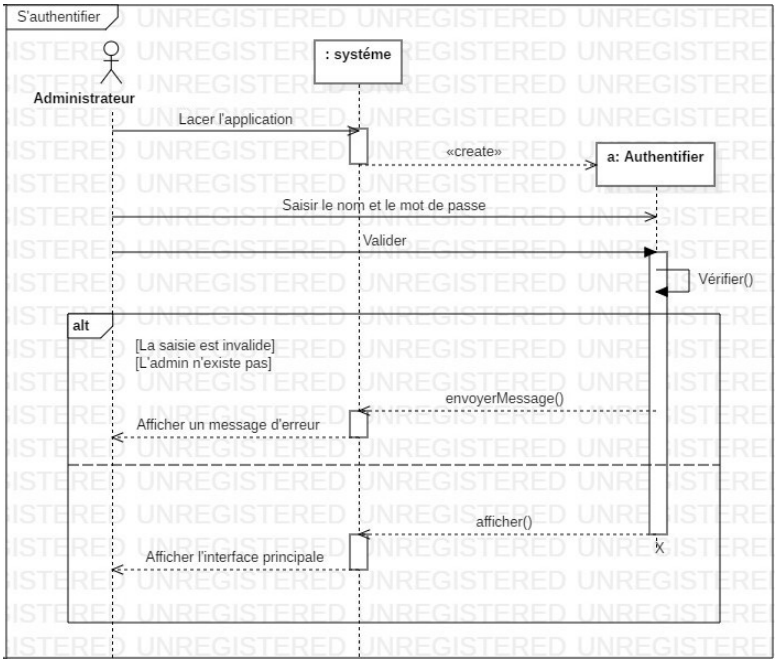

FIGURE 3.4 – Diagramme de séquence cas d'utilisation ≪ S'authentifier ≫.

• **Cas d'utilisation ≪ Introduire l'adresse IP du contrôleur ≫ :**

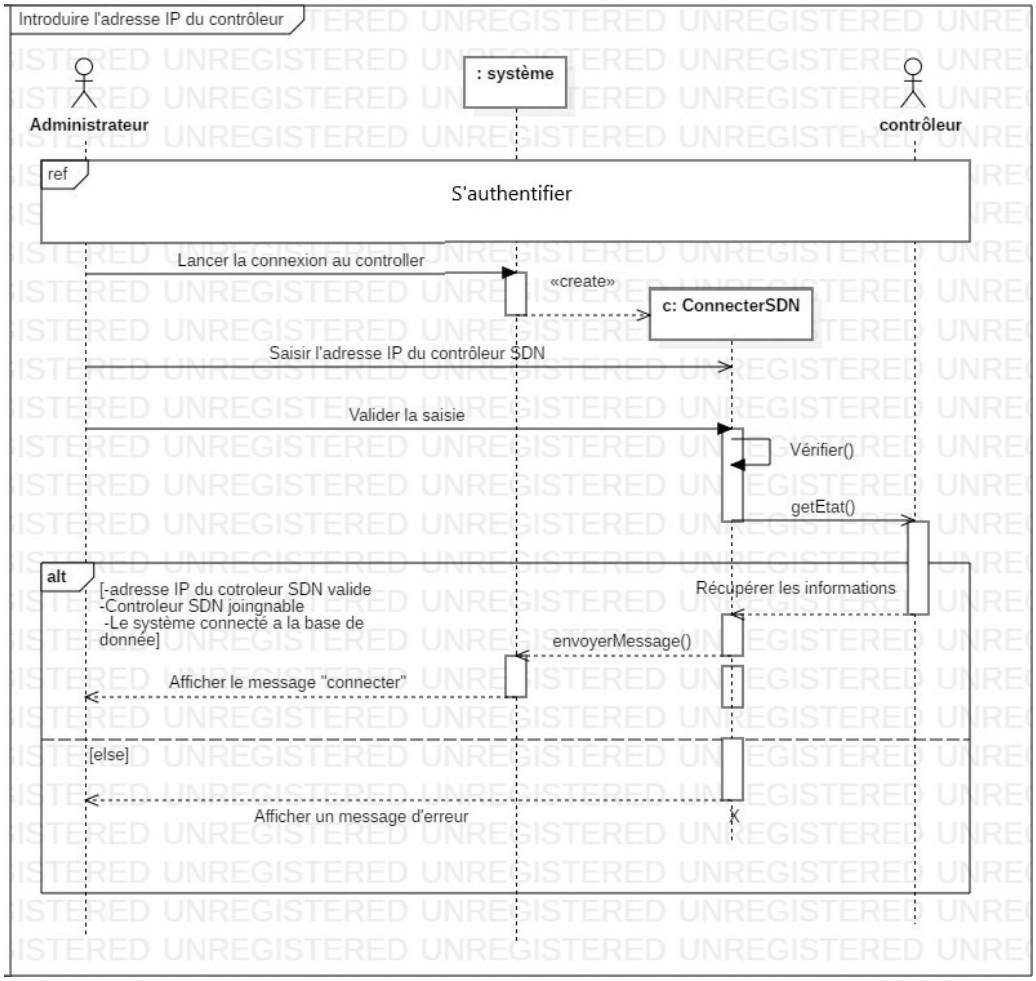

FIGURE 3.5 – Diagramme de séquence cas d'utilisation ≪ Introduire l'adresse IP du contrôleur ≫.

• **Cas d'utilisation « Introduire le préfixe réseau IPv6 global » :**

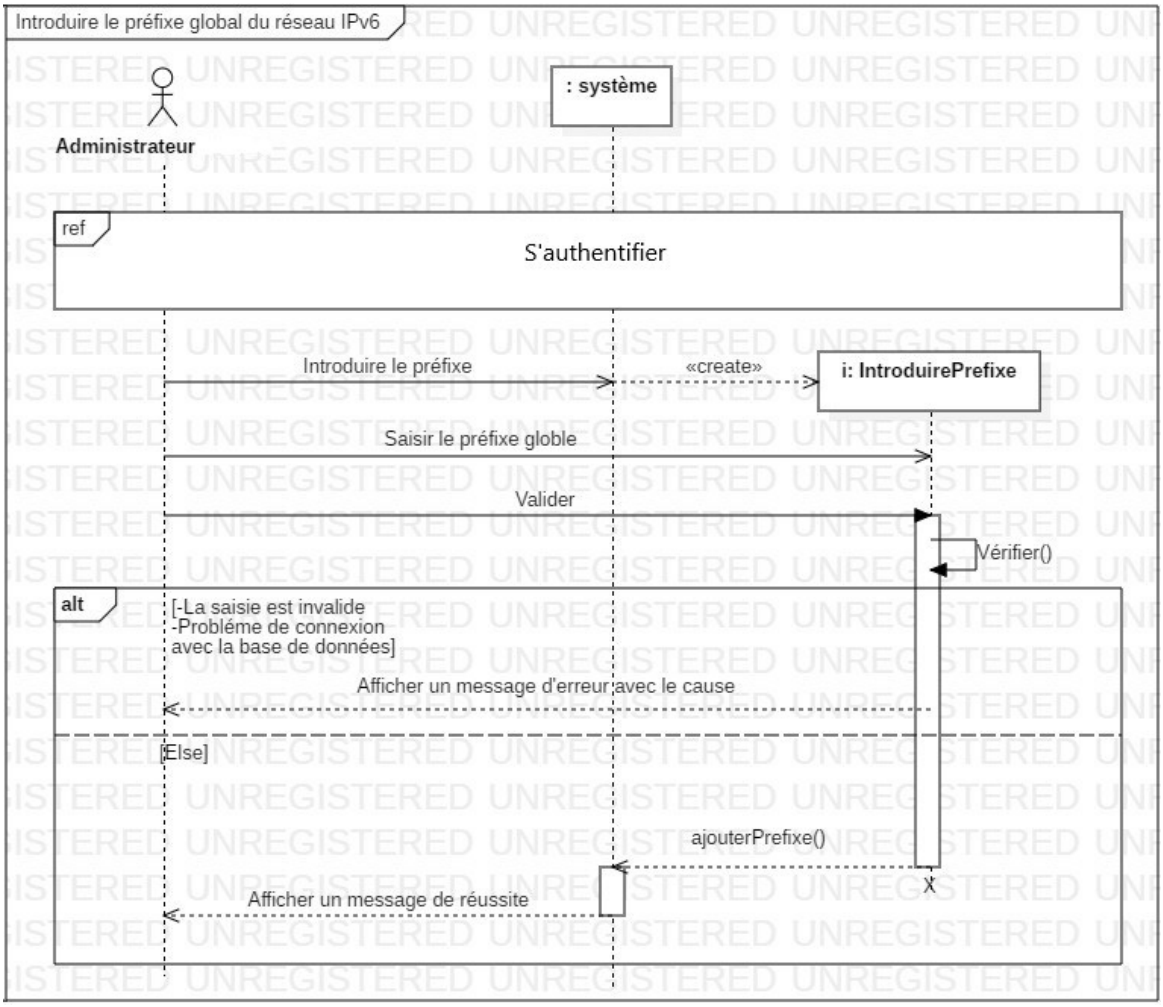

FIGURE 3.6 – Diagramme de séquence cas d'utilisation « Introduire le préfixe réseau IPv6 global ».

• **Cas d'utilisation « Ajouter un niveau » :**

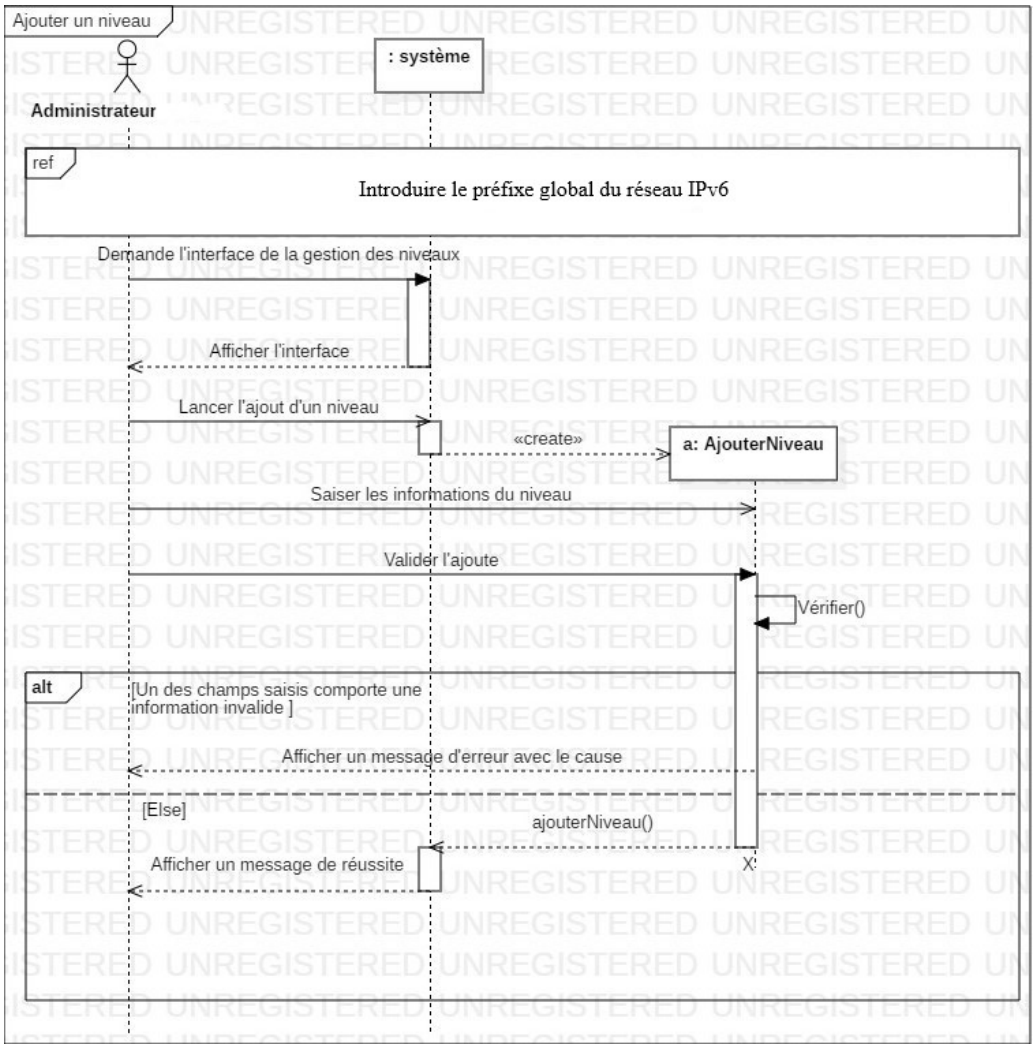

FIGURE 3.7 – Diagramme de séquence cas d'utilisation « Ajouter un niveau ».

• **Cas d'utilisation ≪ Modifier un niveau ≫ :**

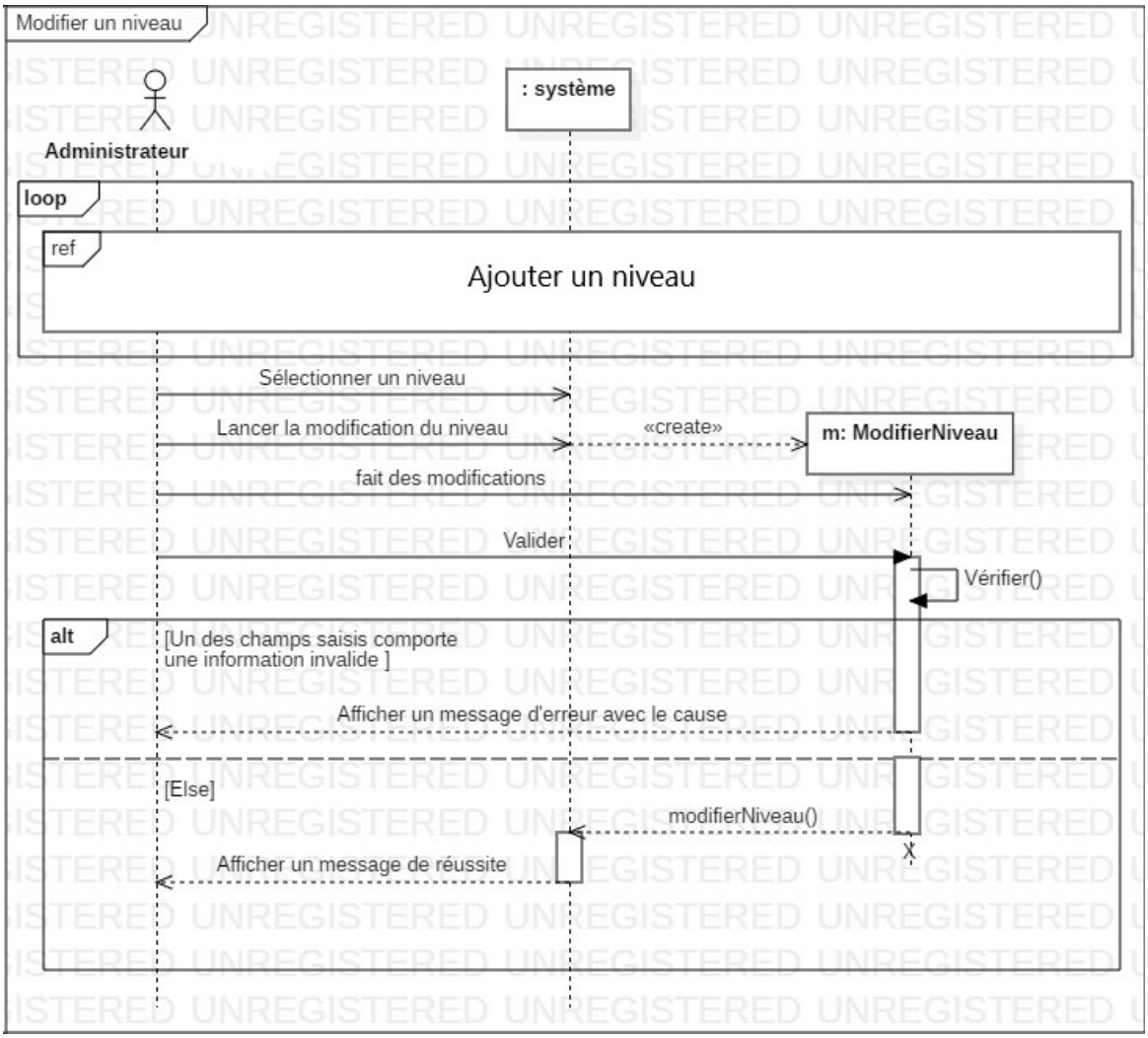

FIGURE 3.8 – Diagramme de séquence cas d'utilisation ≪ Modifier un niveau ≫.

• **Cas d'utilisation ≪ Supprimer un niveau ≫ :**

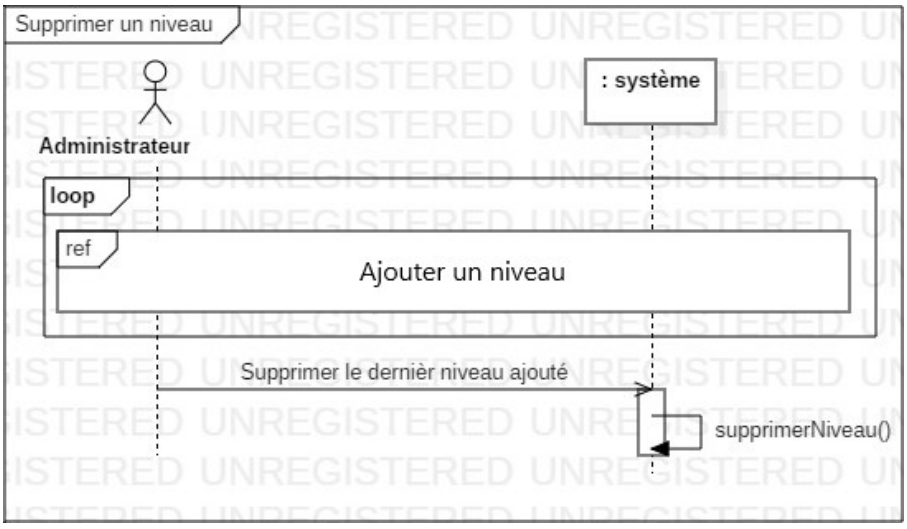

FIGURE 3.9 – Diagramme de séquence cas d'utilisation ≪ Supprimer un niveau ≫

• **Cas d'utilisation ≪ MAJ les niveaux hiérarchiques ≫ :**

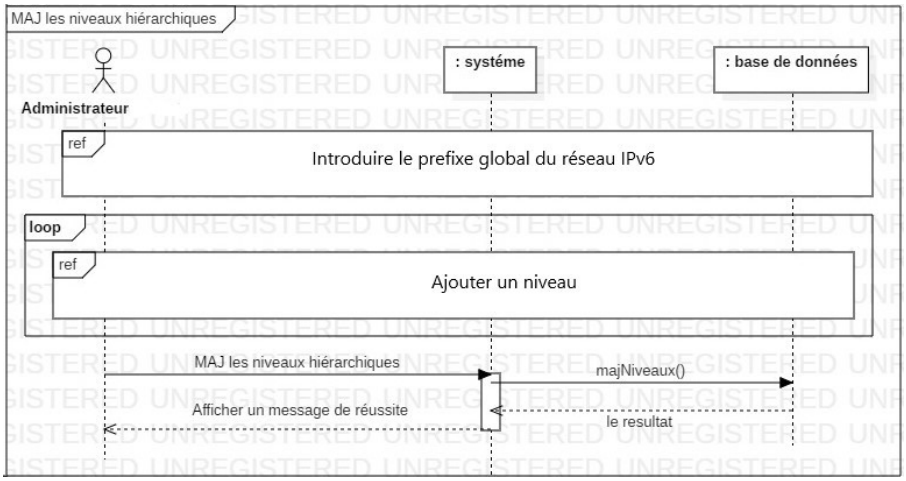

FIGURE 3.10 – Diagramme de séquence cas d'utilisation ≪ MAJ les niveaux hiérarchiques ≫.

• **Cas d'utilisation ≪ Ajouter une structure ≫ :**

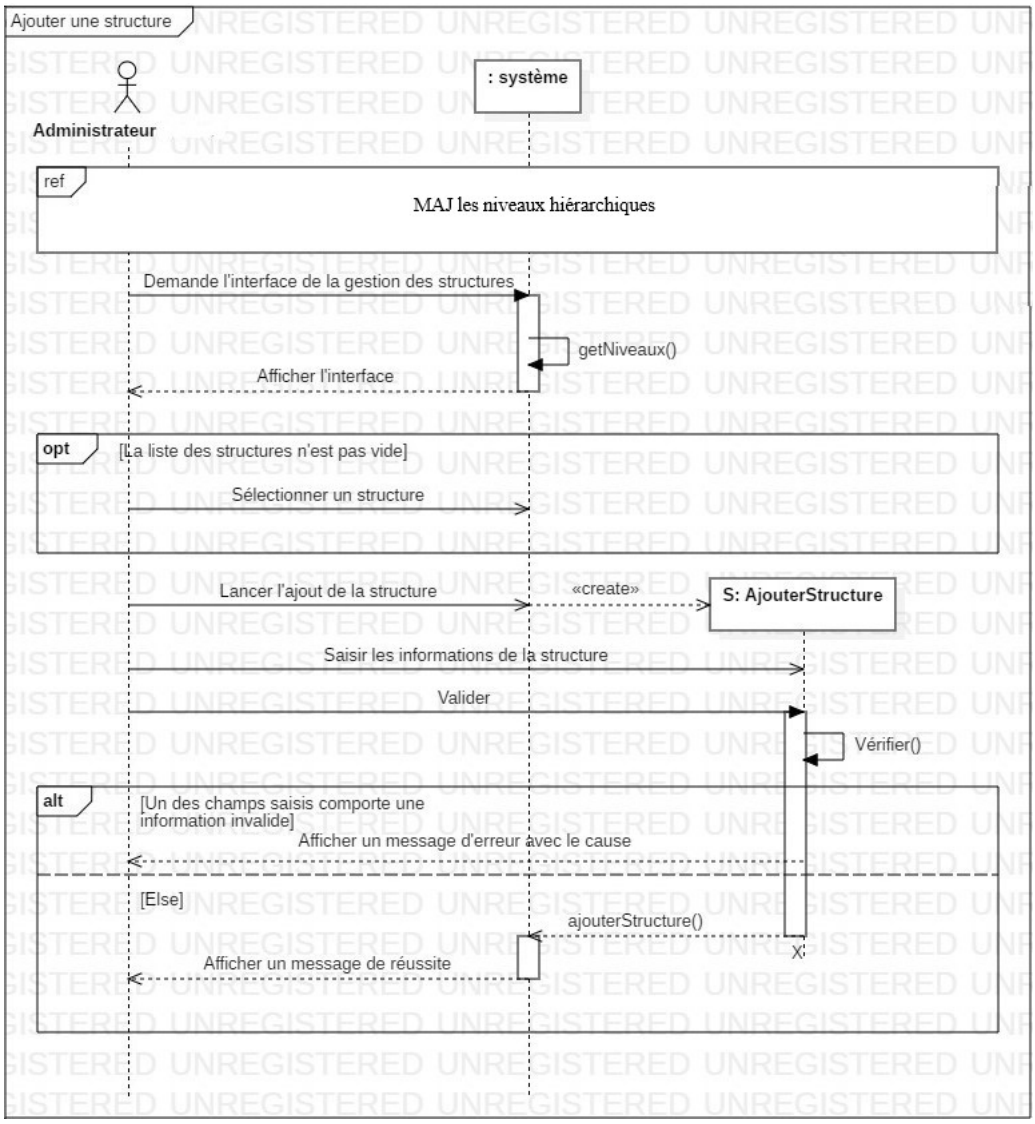

FIGURE 3.11 – Diagramme de séquence cas d'utilisation ≪ Ajouter une structure ≫.

• **Cas d'utilisation ≪ Modifier une structure ≫ :**

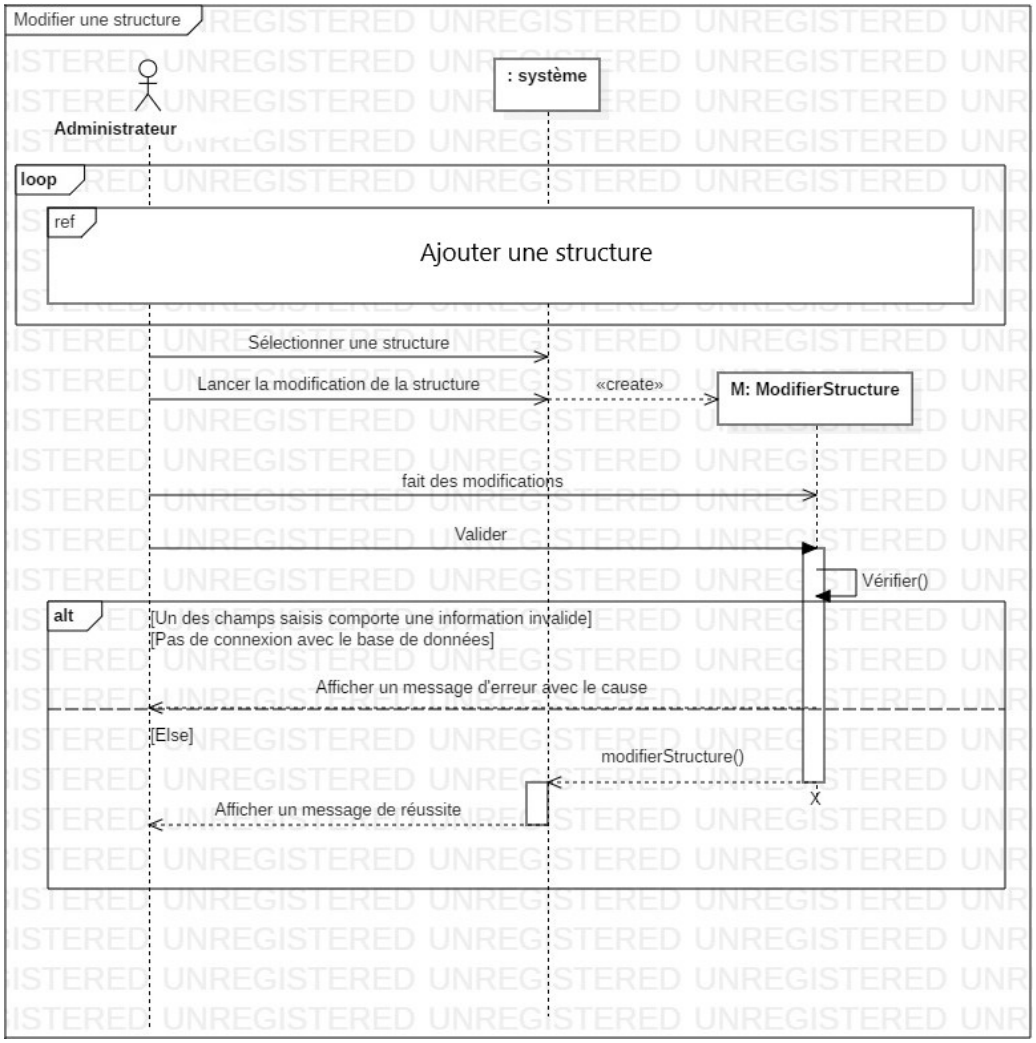

FIGURE 3.12 – Diagramme de séquence cas d'utilisation ≪ Modifier une structure ≫.

• **Cas d'utilisation ≪ Supprimer une structure ≫ :**

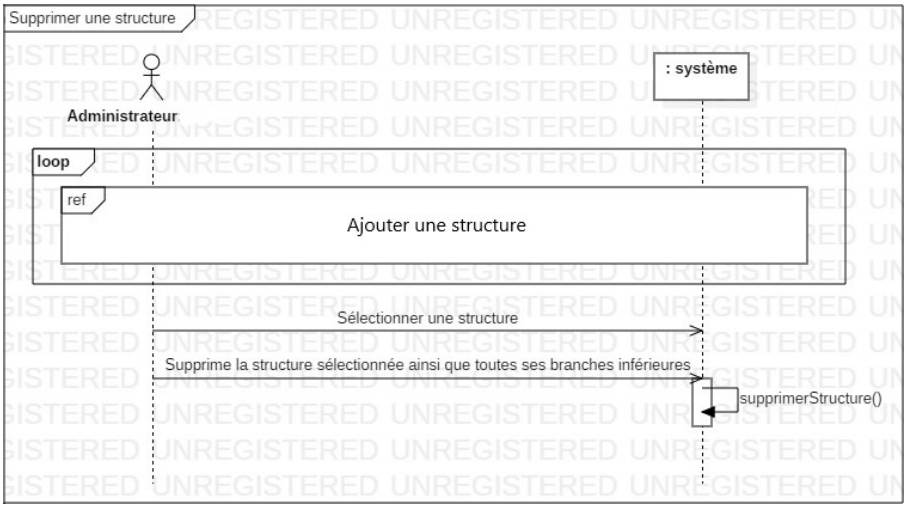

FIGURE 3.13 – Diagramme de séquence cas d'utilisation ≪ Supprimer une structure ≫.

• **Cas d'utilisation ≪ MAJ les structures organisationnelles ≫ :**

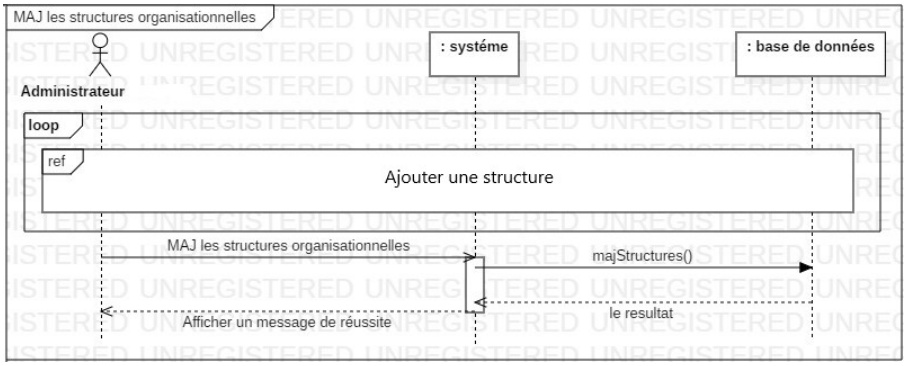

FIGURE 3.14 – Diagramme de séquence cas d'utilisation ≪ MAJ les structures organisationnelles ≫.

• **Cas d'utilisation « Ajouter une politique » :**

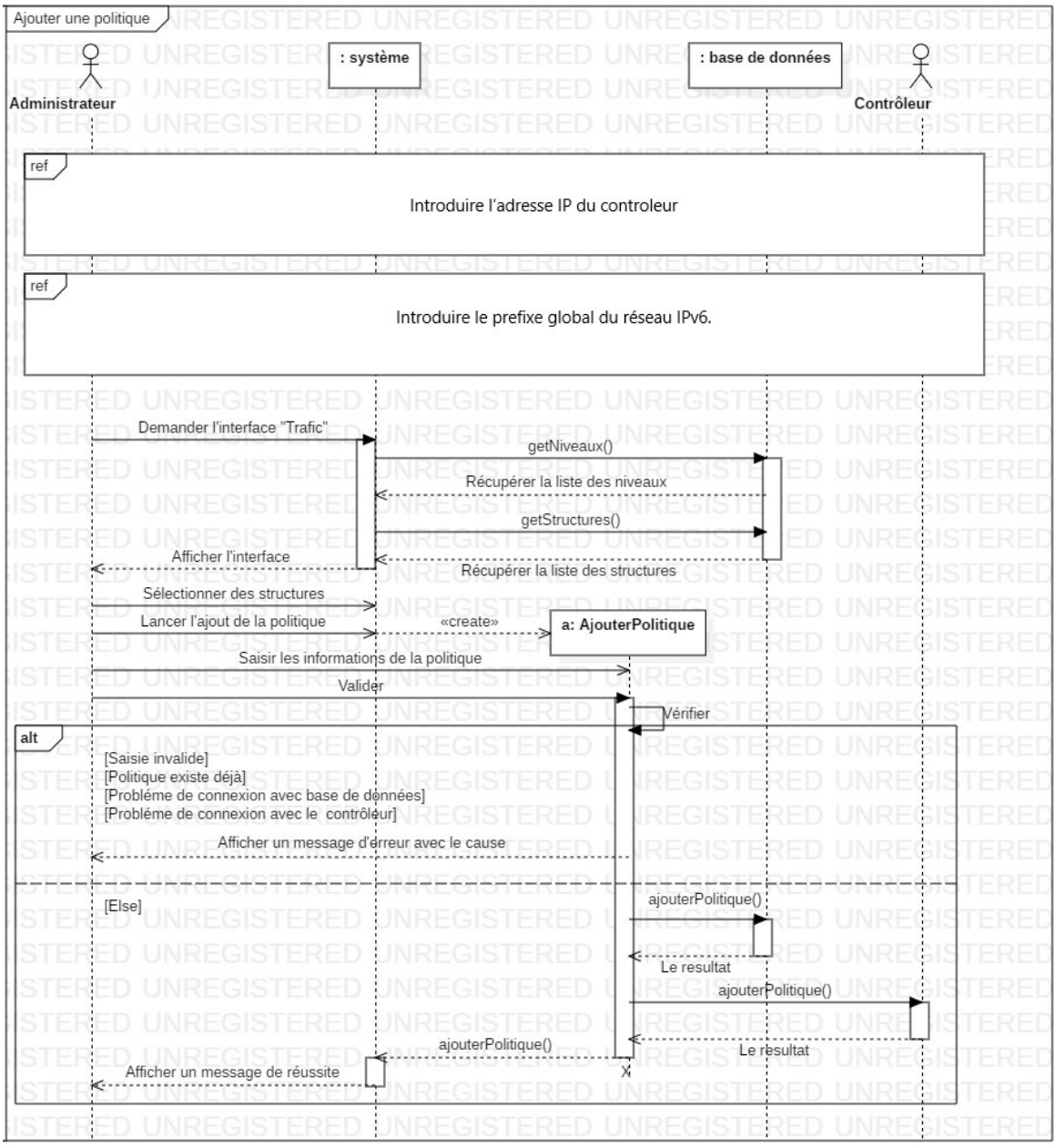

FIGURE 3.15 – Diagramme de séquence cas d'utilisation « Ajouter une politique ».

• **Cas d'utilisation « Afficher les politiques » :**

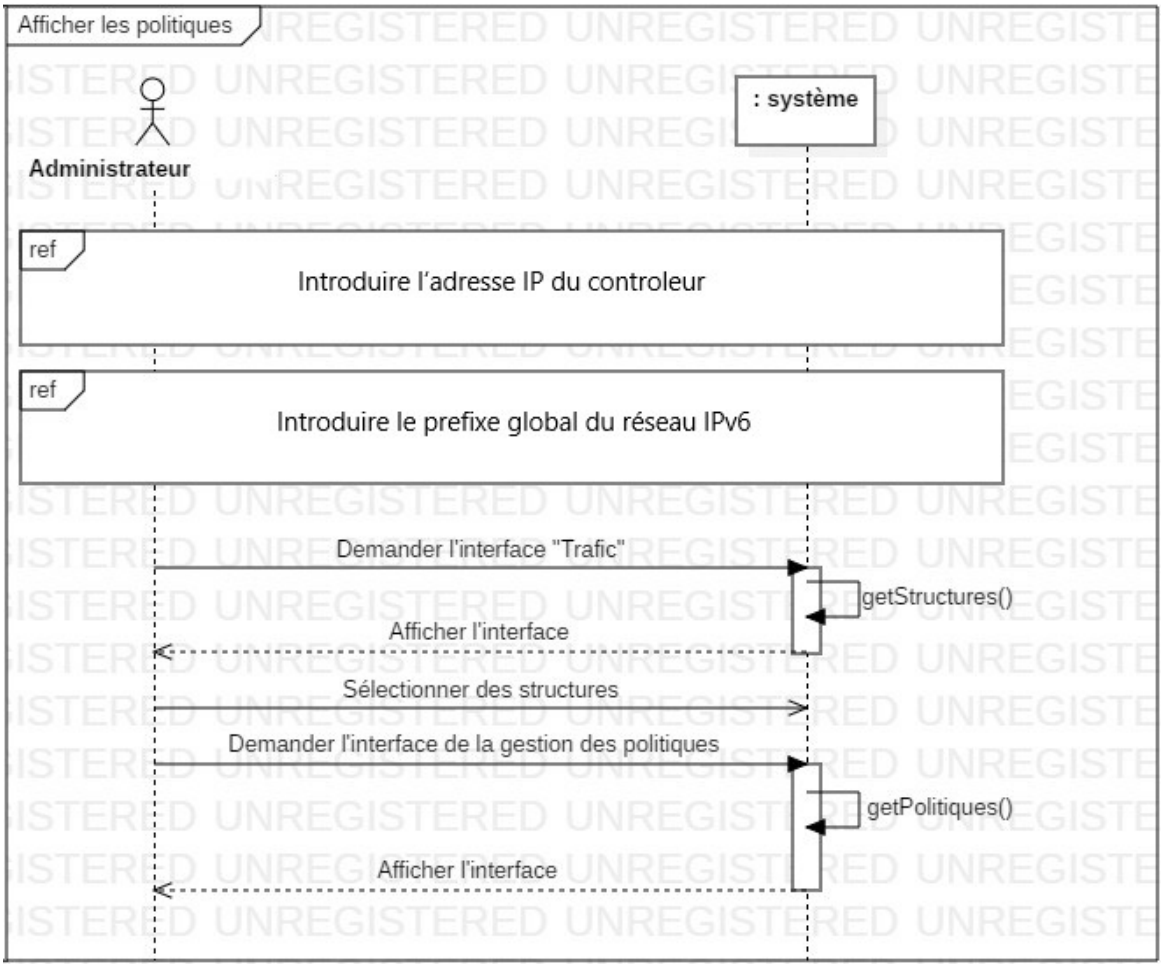

FIGURE 3.16 – Diagramme de séquence cas d'utilisation « Afficher les politiques ».

• **Cas d'utilisation ≪ Modifier une politique ≫ :**

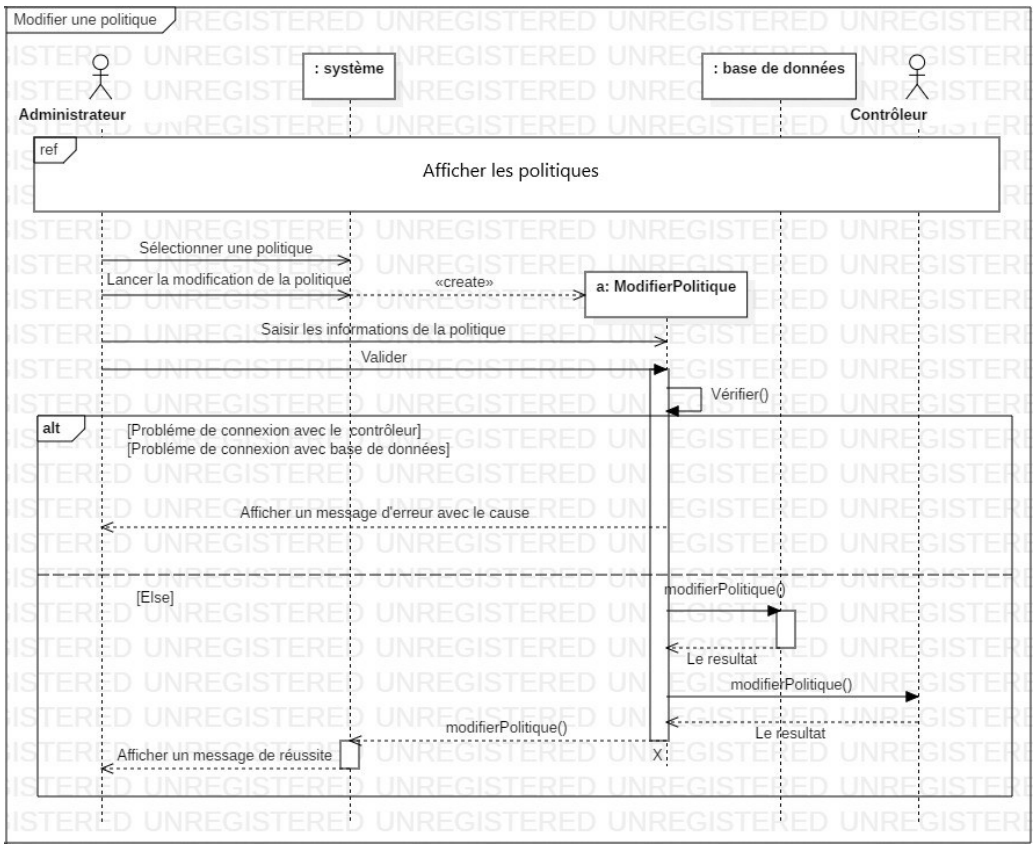

FIGURE 3.17 – Diagramme de séquence cas d'utilisation ≪ Modifier une politique ≫.

• **Cas d'utilisation ≪ Supprimer une politique ≫ :**

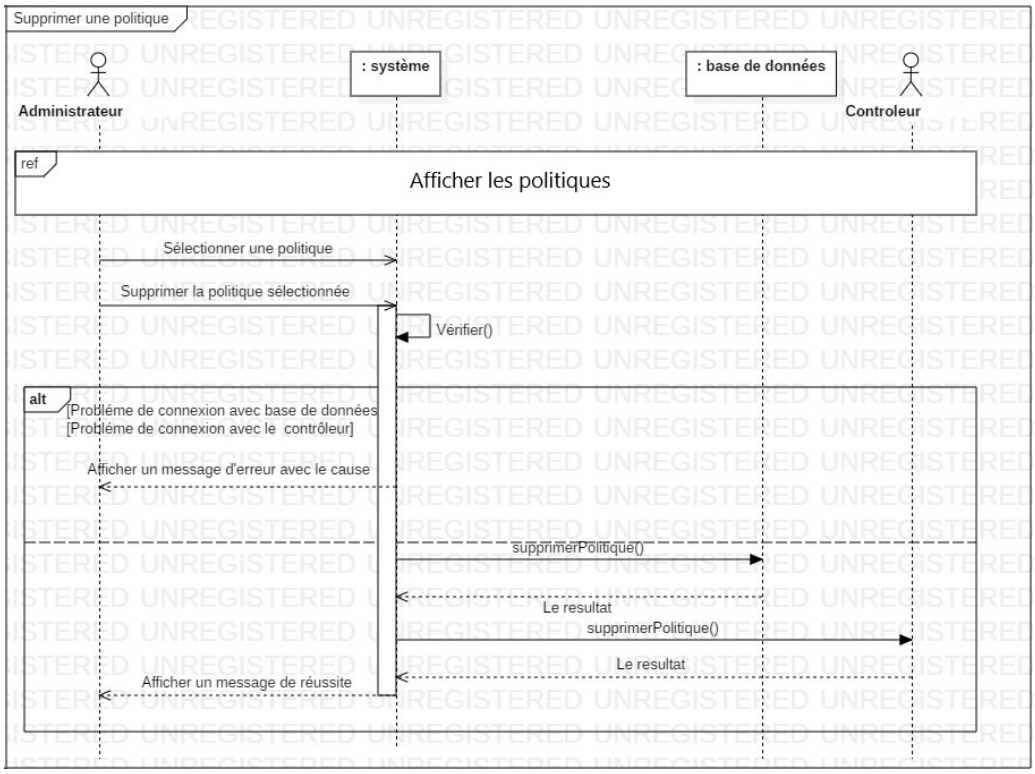

FIGURE 3.18 – Diagramme de séquence cas d'utilisation ≪ Supprimer une politique ≫.

• **Cas d'utilisation ≪ Réinitialiser les politiques ≫ :**

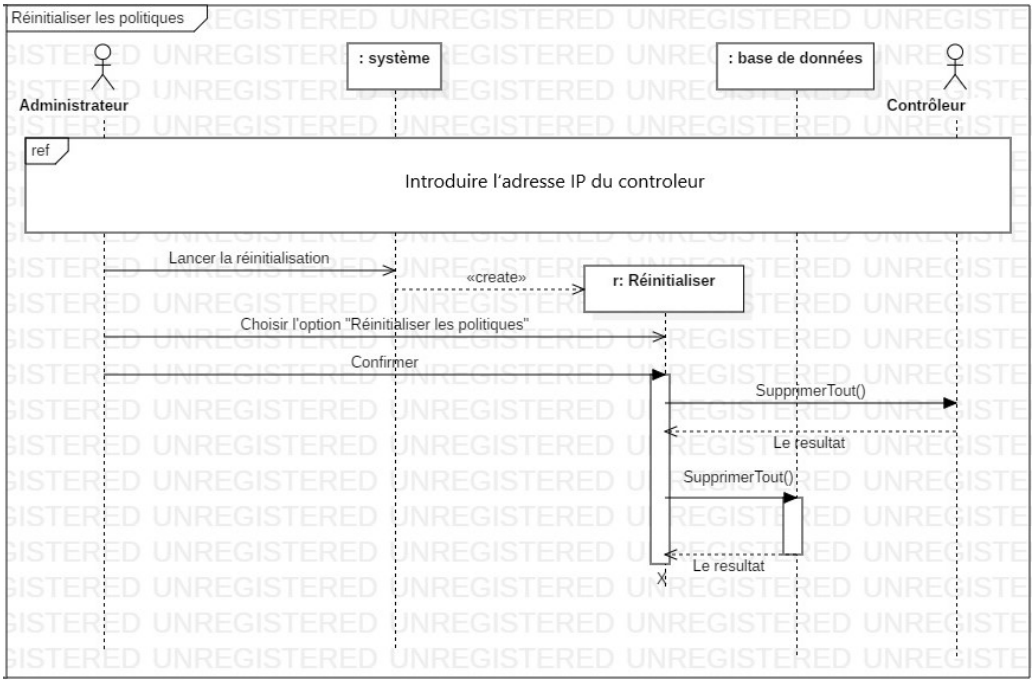

FIGURE 3.19 – Diagramme de séquence cas d'utilisation ≪ Réinitialiser les politiques ≫.

• **Cas d'utilisation ≪ Réinitialiser le réseau ≫ :**

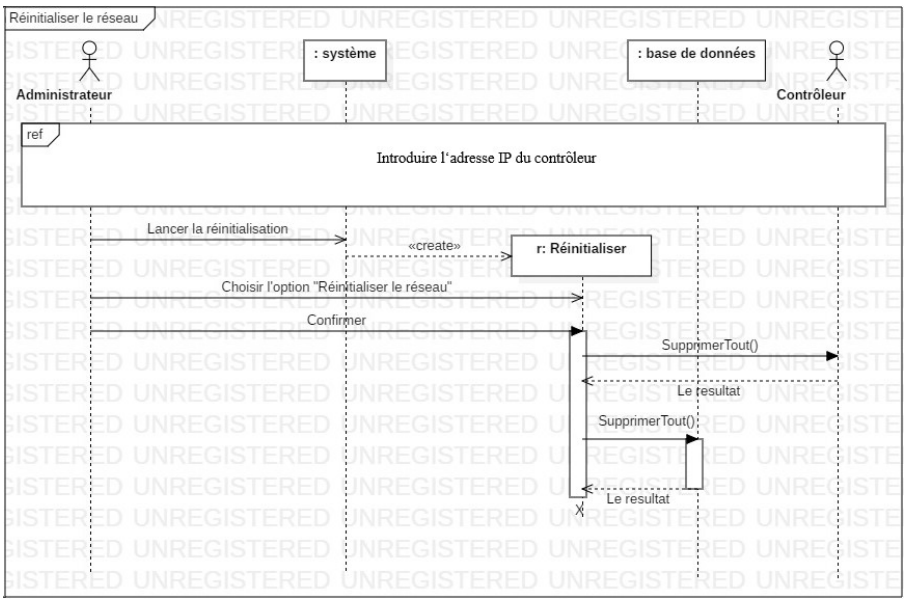

FIGURE 3.20 – Diagramme de séquence cas d'utilisation ≪ Réinitialiser le réseau ≫.

• **Cas d'utilisation ≪ Chercher des politique ≫ :**

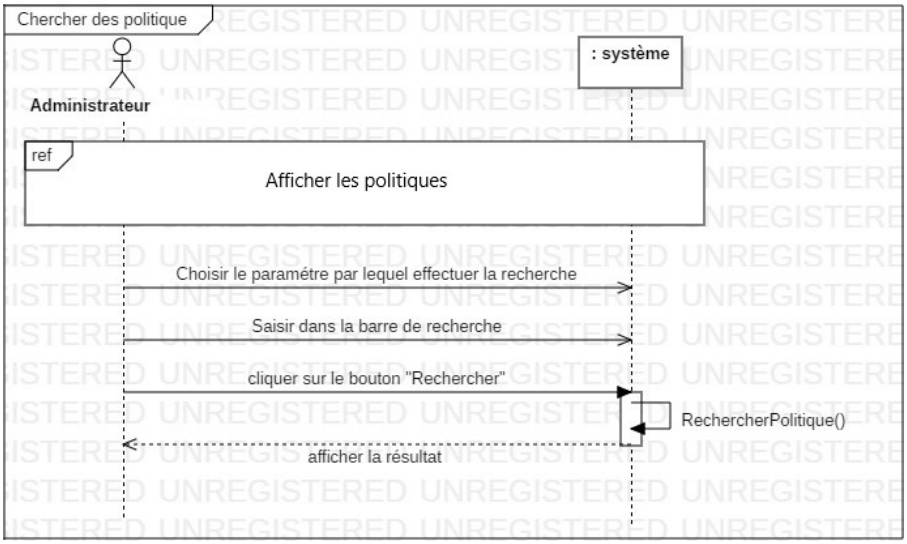

FIGURE 3.21 – Diagramme de séquence cas d'utilisation ≪ Chercher des politique ≫.

B-Phase d'analyse :

La phase d'analyse permet de décrire le comportement du système :

• Diagramme d'activités :

• Définition :

Le diagramme d'activité représente le déroulement d'un cas d'utilisation réalisé par le système, avec tous les branchements conditionnels et toutes les boucles possibles.

Ces diagrammes permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation [58].

• Les composants de base du diagramme d'activités :

— **Nœud initial :** Il indique le début du déroulement d'un cas d'utilisation modélisé. Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Graphiquement, un nœud initial est représenté par un petit cercle plein.

— **Nœud final :** Il indique la fin du déroulement d'un cas d'utilisation modélisé. Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant. Graphiquement, un nœud final est représenté par un cercle plein entouré d'un autre cercle.

— **Nœud de décision :** Un nœud de décision est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. Ces derniers sont généralement accompagnés de conditions de garde pour conditionner le choix. Graphiquement, on représente un nœud de décision par un losange.

— **Le nœud d'action :** Un nœud d'action est un état d'activité exécutable qui constitue l'unité fondamentale de fonctionnalité exécutable dans une activité.

— **La transition :** Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage. Les transitions sont modélisées par des flèches.

• **Cas d'utilisation « S'authentifier »**

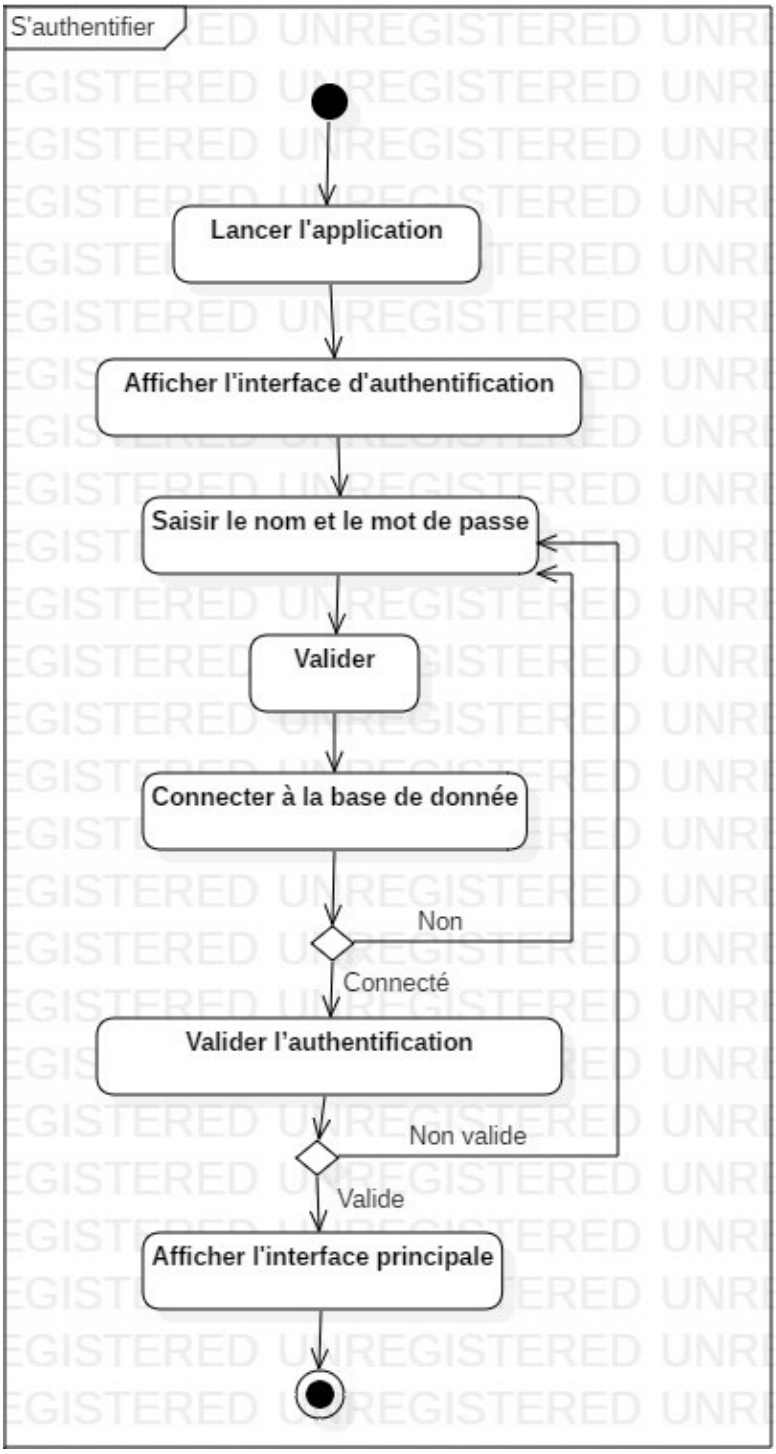

FIGURE 3.22 – Diagramme d'activité du cas d'utilisation « S'authentifier ».

• **Cas d'utilisation ≪ Introduire l'adresse IP du contrôleur ≫ :**

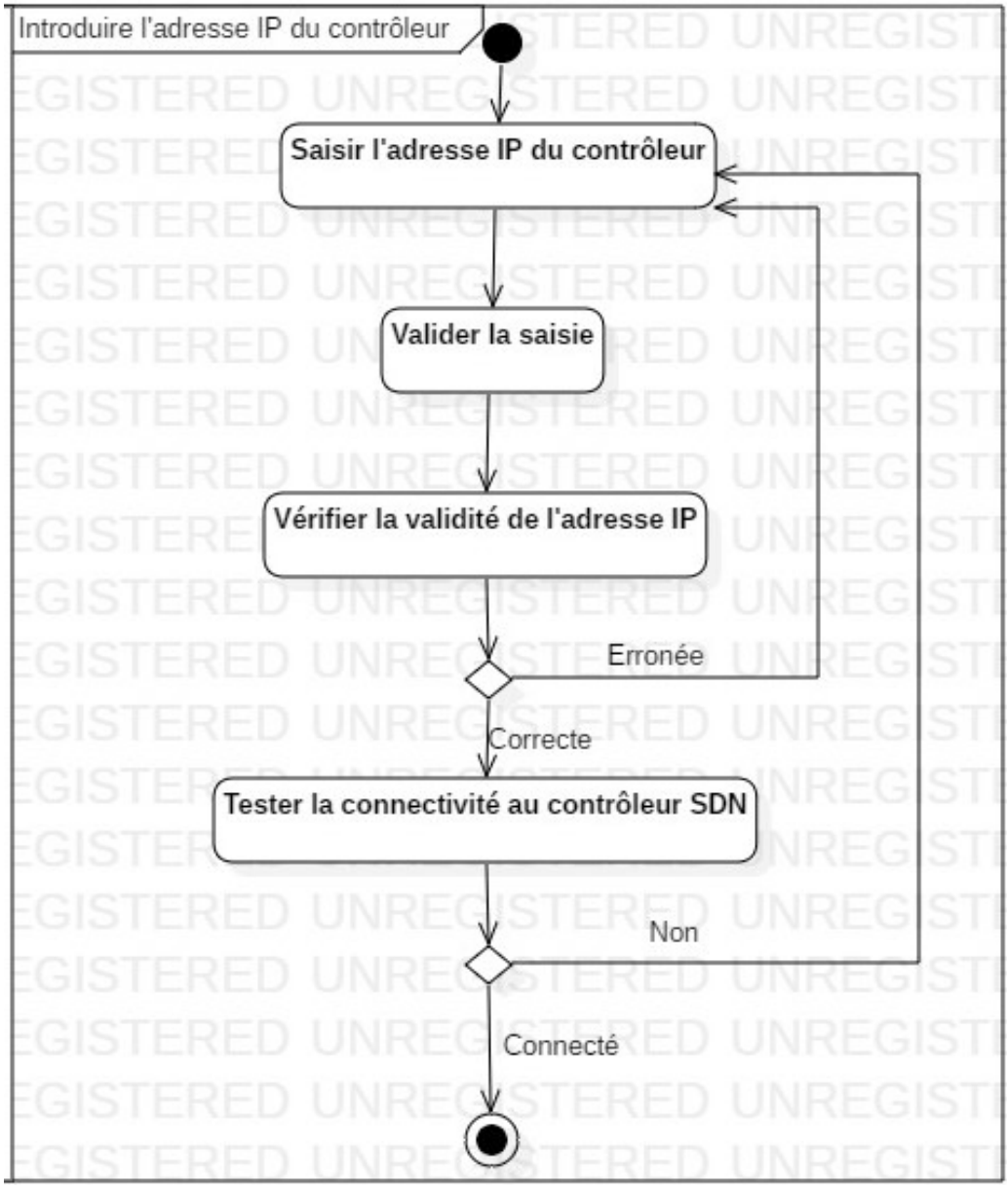

FIGURE 3.23 – Diagramme d'activité du cas d'utilisation ≪ Introduire l'adresse IP du contrôleur ≫.

• **Cas d'utilisation « Introduire le préfixe global du réseau IPv6 » :**

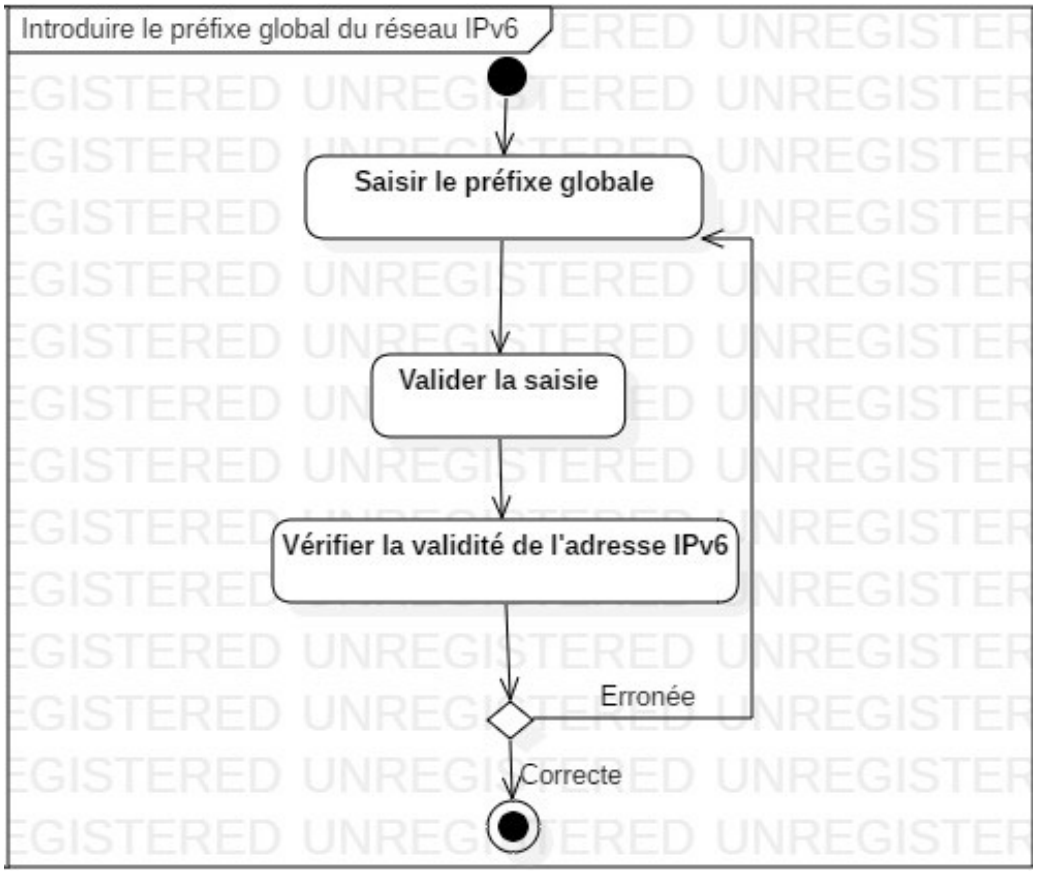

FIGURE 3.24 – Diagramme d'activité du cas d'utilisation «Introduire le préfixe global du réseau IPv6».

• **Cas d'utilisation « Ajouter un niveau » :**

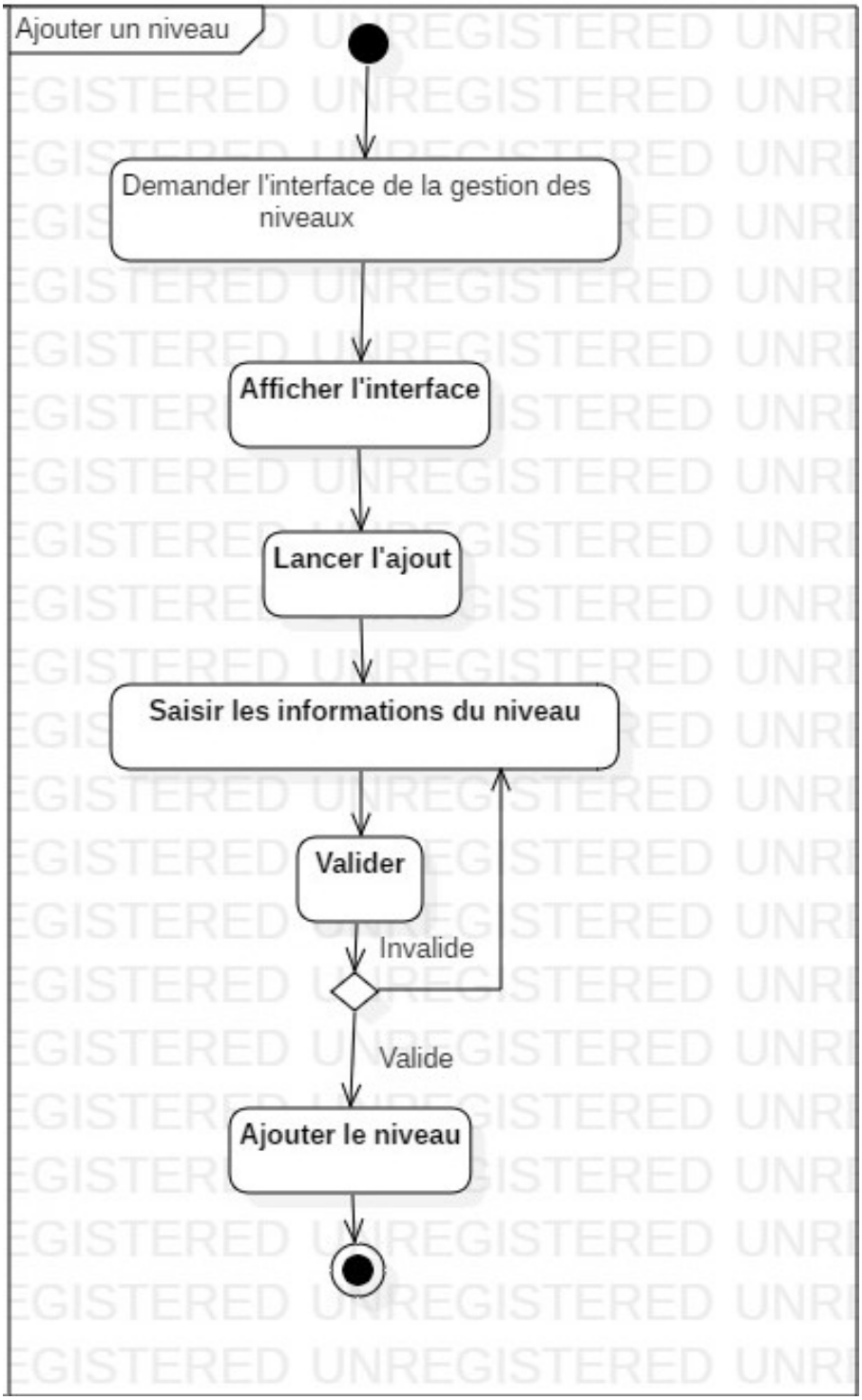

FIGURE 3.25 – Diagramme d'activité du cas d'utilisation « Ajouter un niveau ».

• **Cas d'utilisation ≪ Modifier un niveau ≫ :**

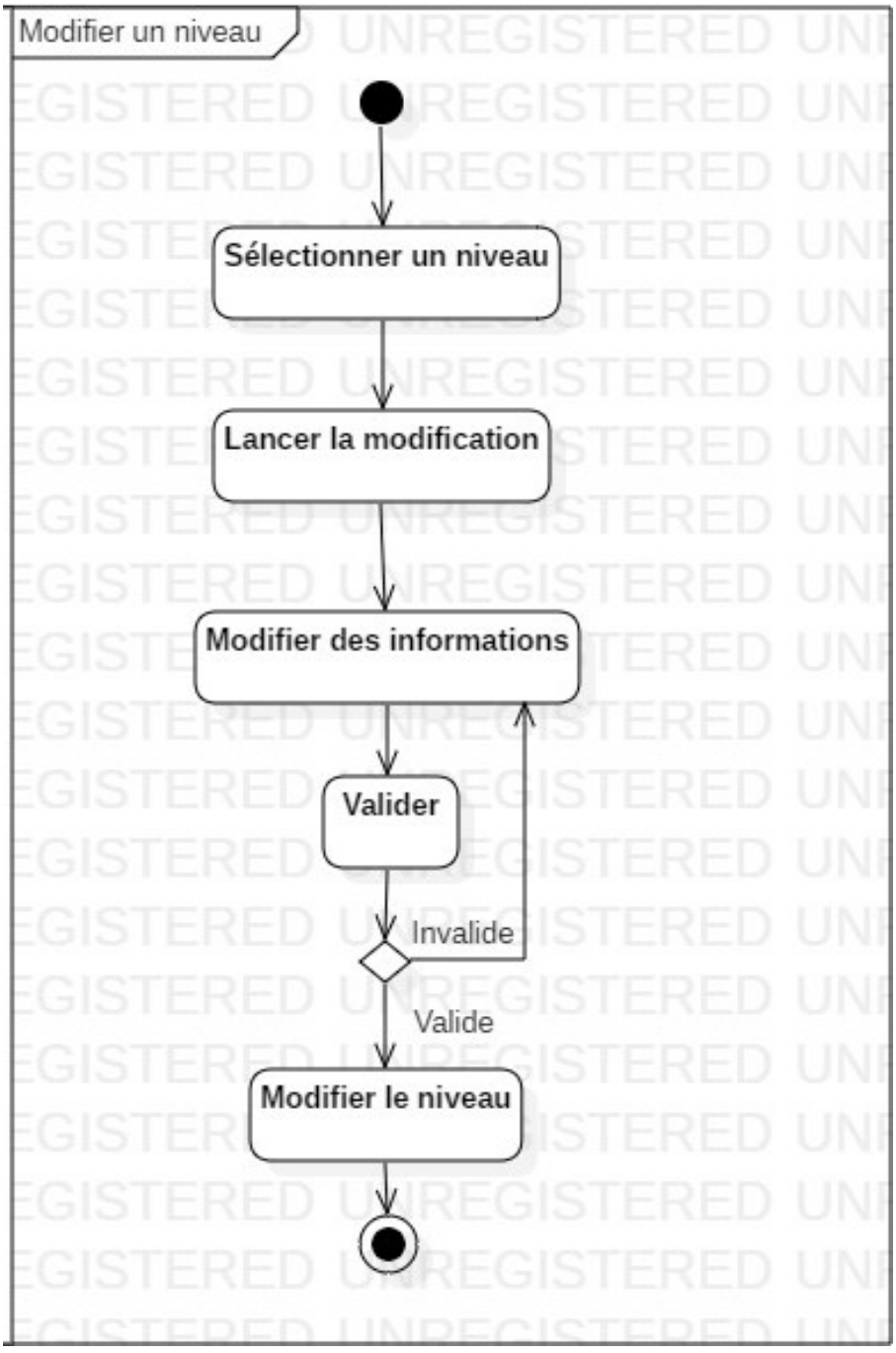

FIGURE 3.26 – Diagramme d'activité du cas d'utilisation ≪ Modifier un niveau ≫.

• **Cas d'utilisation ≪ Ajouter une structure ≫ :**

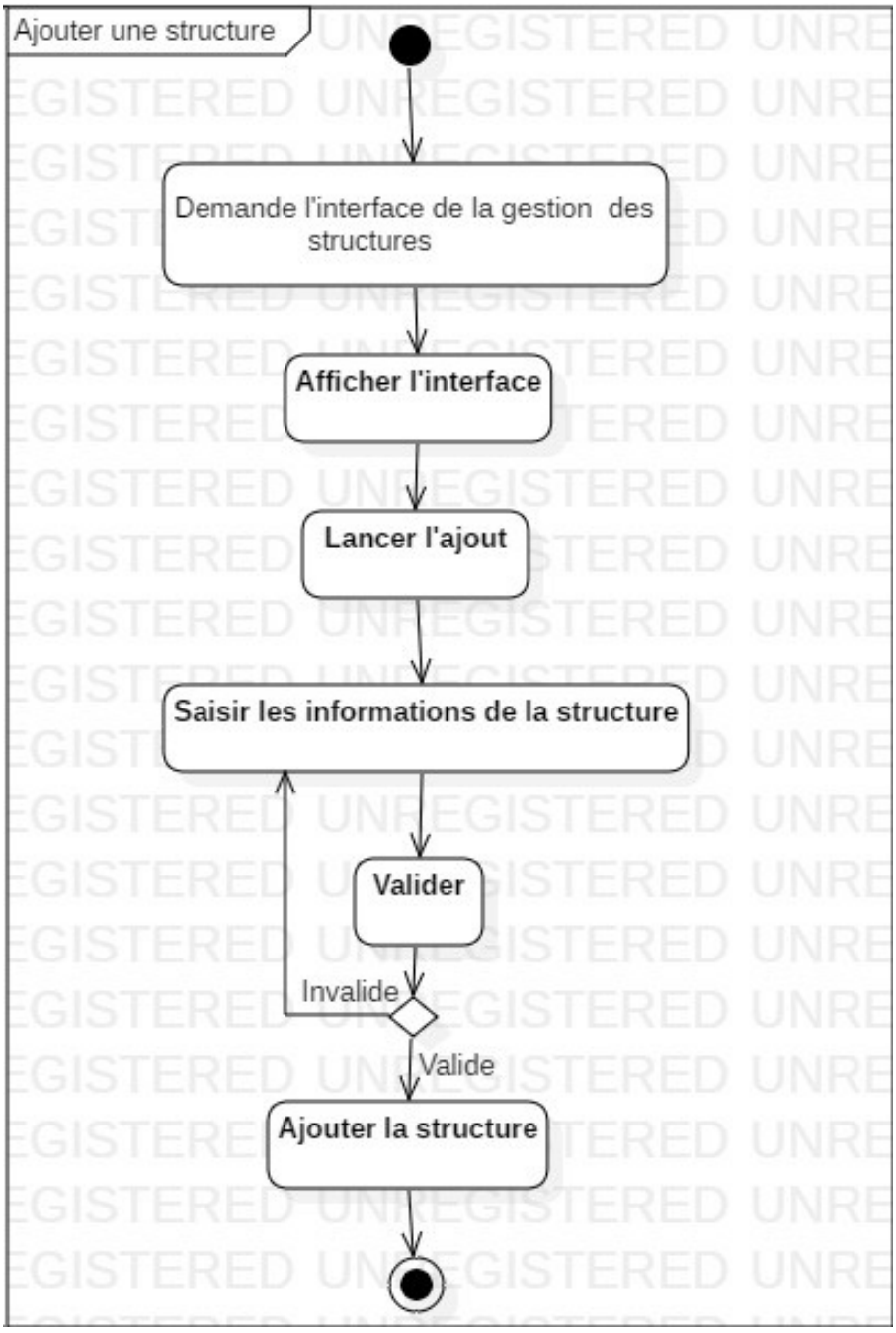

FIGURE 3.27 – Diagramme d'activité du cas d'utilisation ≪ Ajouter une structure ≫.

• **Cas d'utilisation ≪ Modifier une structure ≫ :**

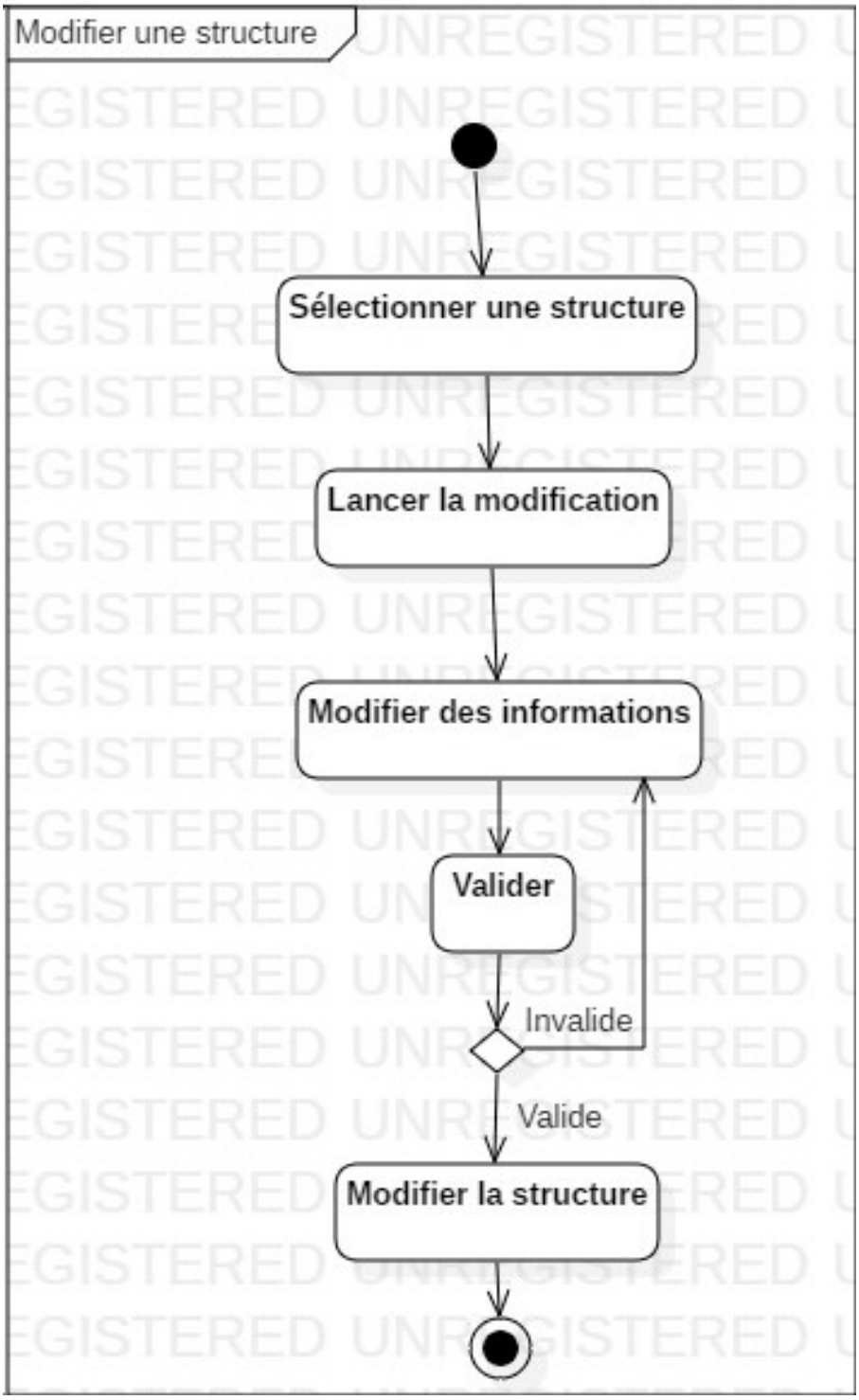

FIGURE 3.28 – Diagramme d'activité du cas d'utilisation ≪ Modifier une structure ≫.

• **Cas d'utilisation ≪ Ajouter une politique ≫ :**

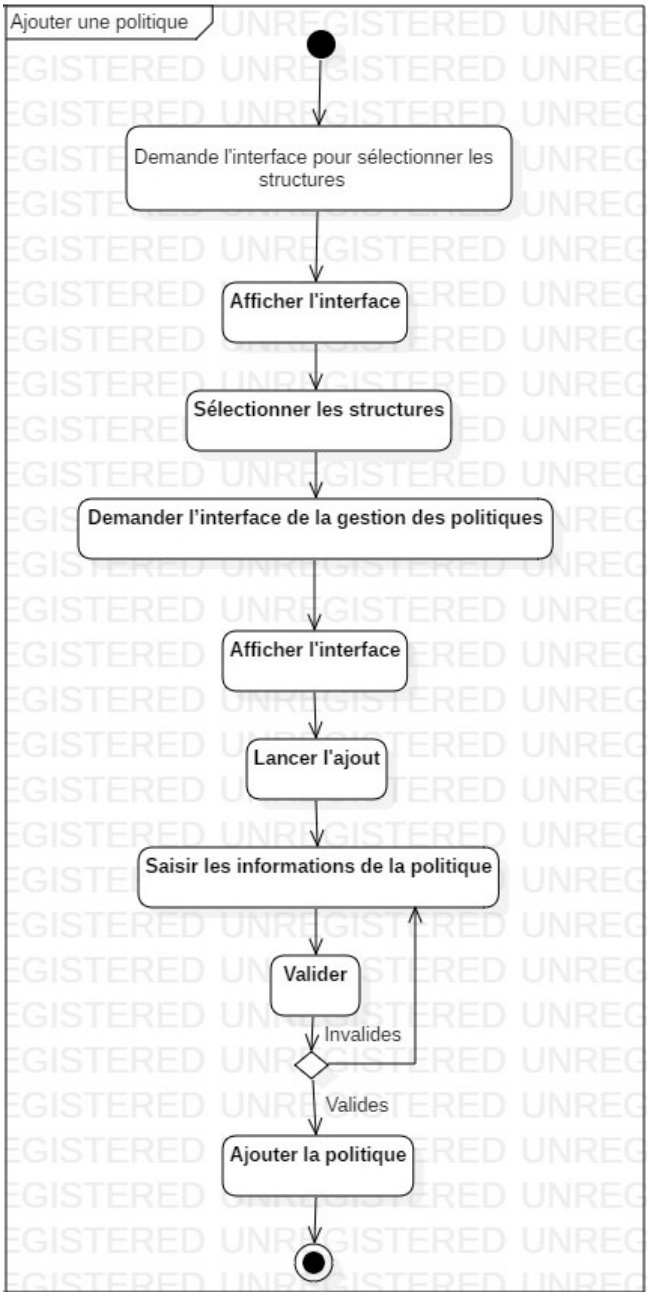

FIGURE 3.29 – Diagramme d'activité du cas d'utilisation ≪ Ajouter une politique ≫.

• **Cas d'utilisation « Afficher les politiques » :**

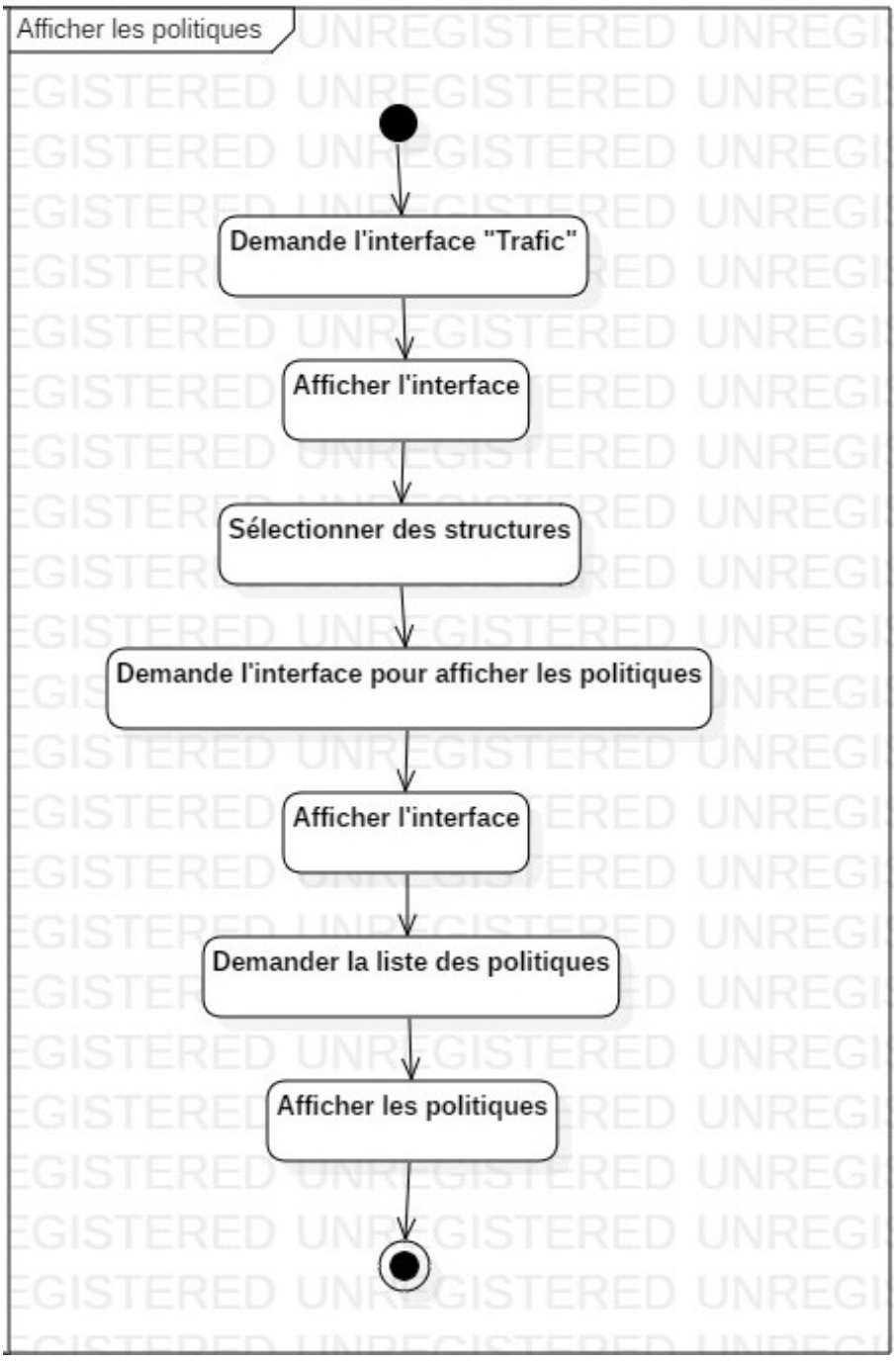

FIGURE 3.30 – Diagramme d'activité du cas d'utilisation « Afficher les politiques ».

• **Cas d'utilisation « Modifier une politique » :**

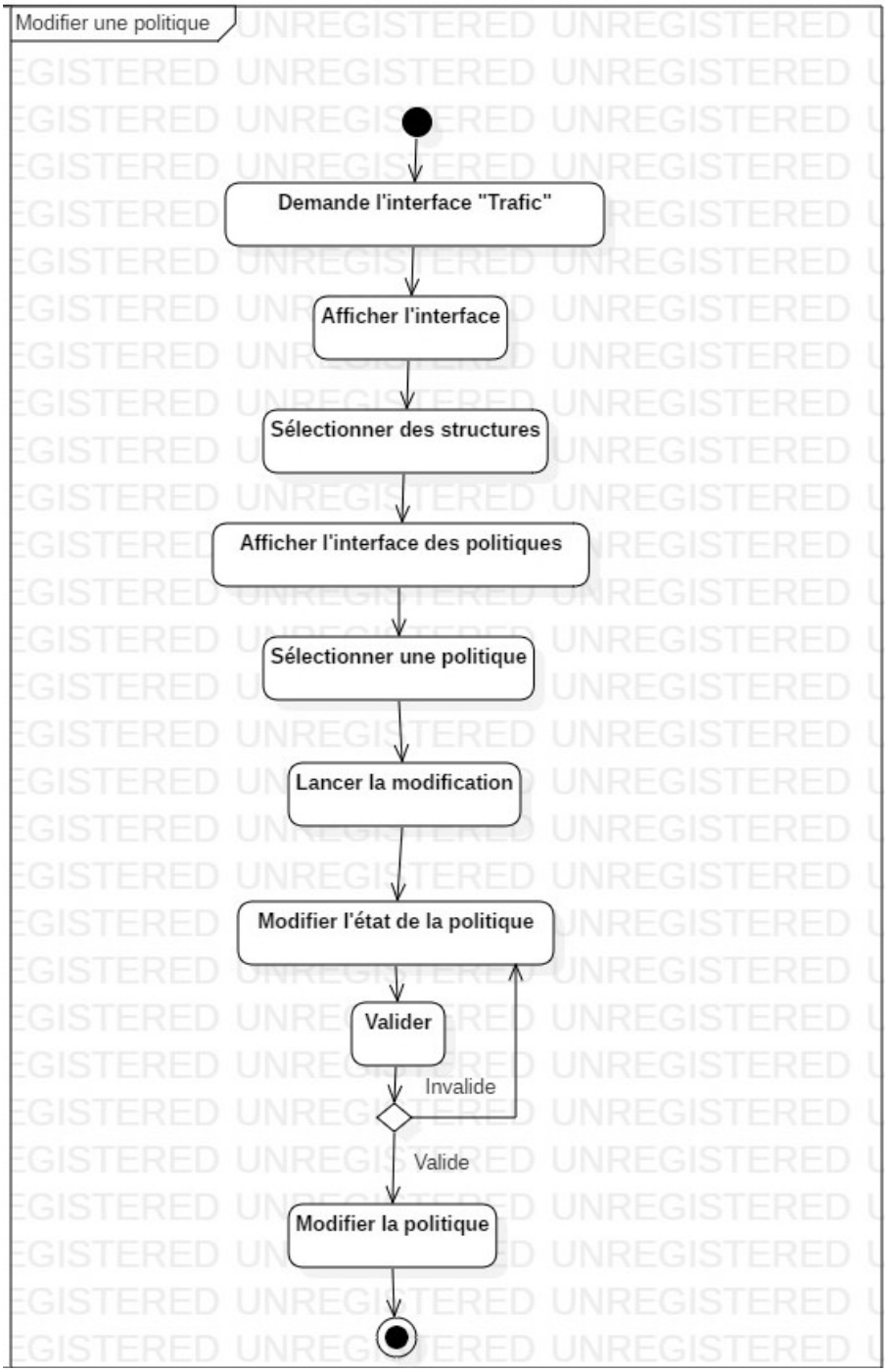

FIGURE 3.31 – Diagramme d'activité du cas d'utilisation « Modifier une politique ».

• **Cas d'utilisation « Supprimer une politique » :**

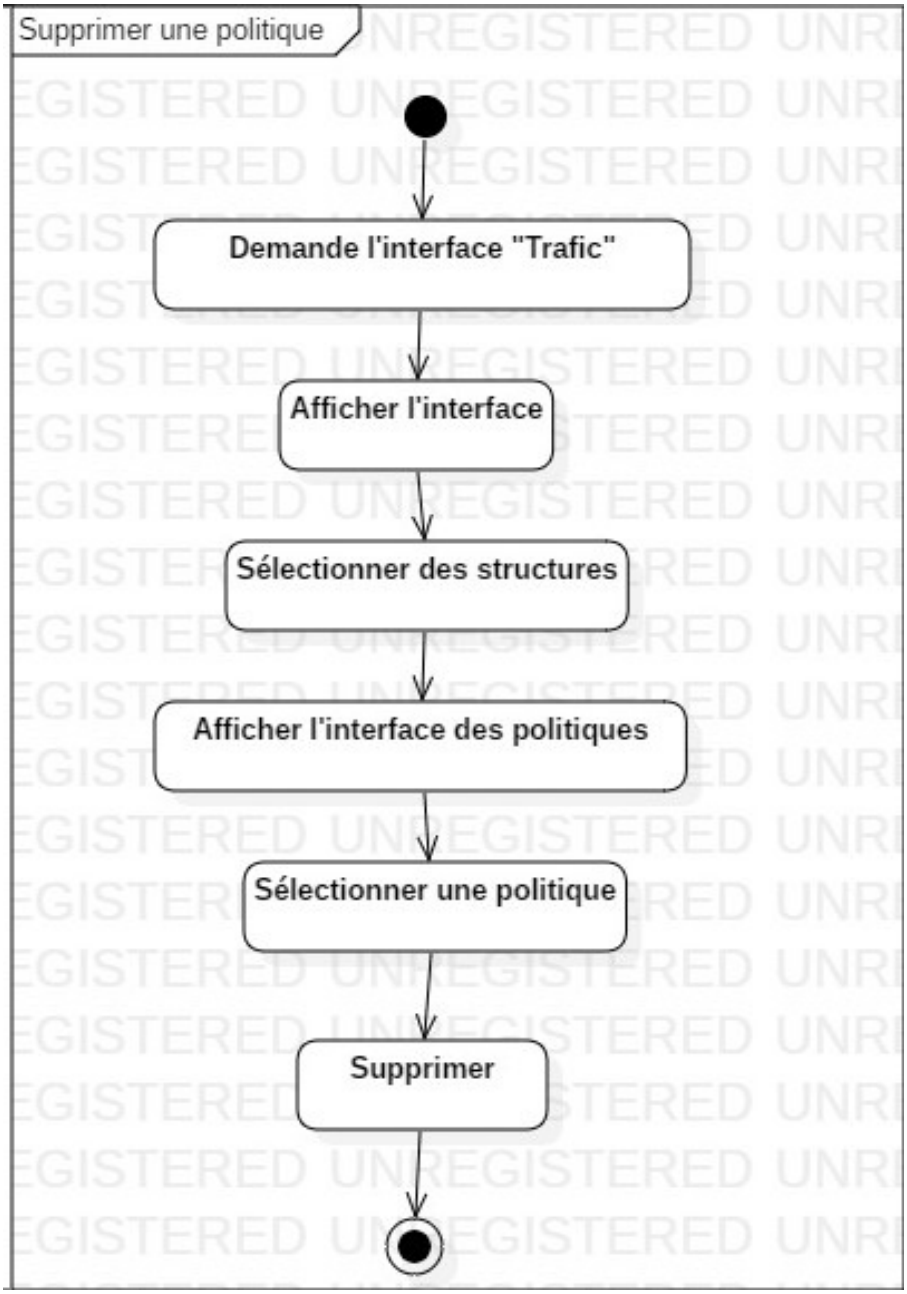

FIGURE 3.32 – Diagramme d'activité du cas d'utilisation « Modifier une politique ».

• **Cas d'utilisation « Réinitialiser les politiques » :**

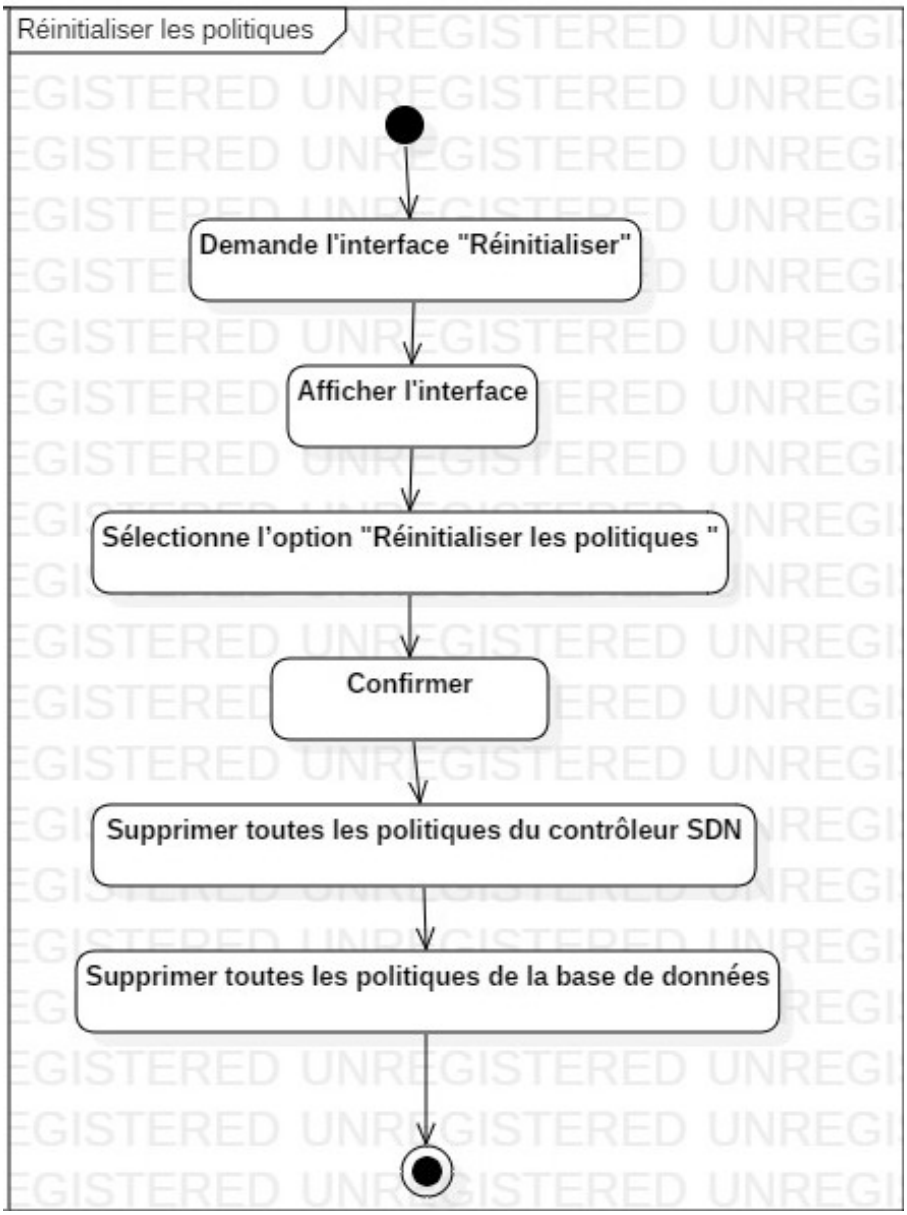

FIGURE 3.33 – Diagramme d'activité du cas d'utilisation « Réinitialiser les politiques ».

• **Cas d'utilisation « Réinitialiser le réseau » :**

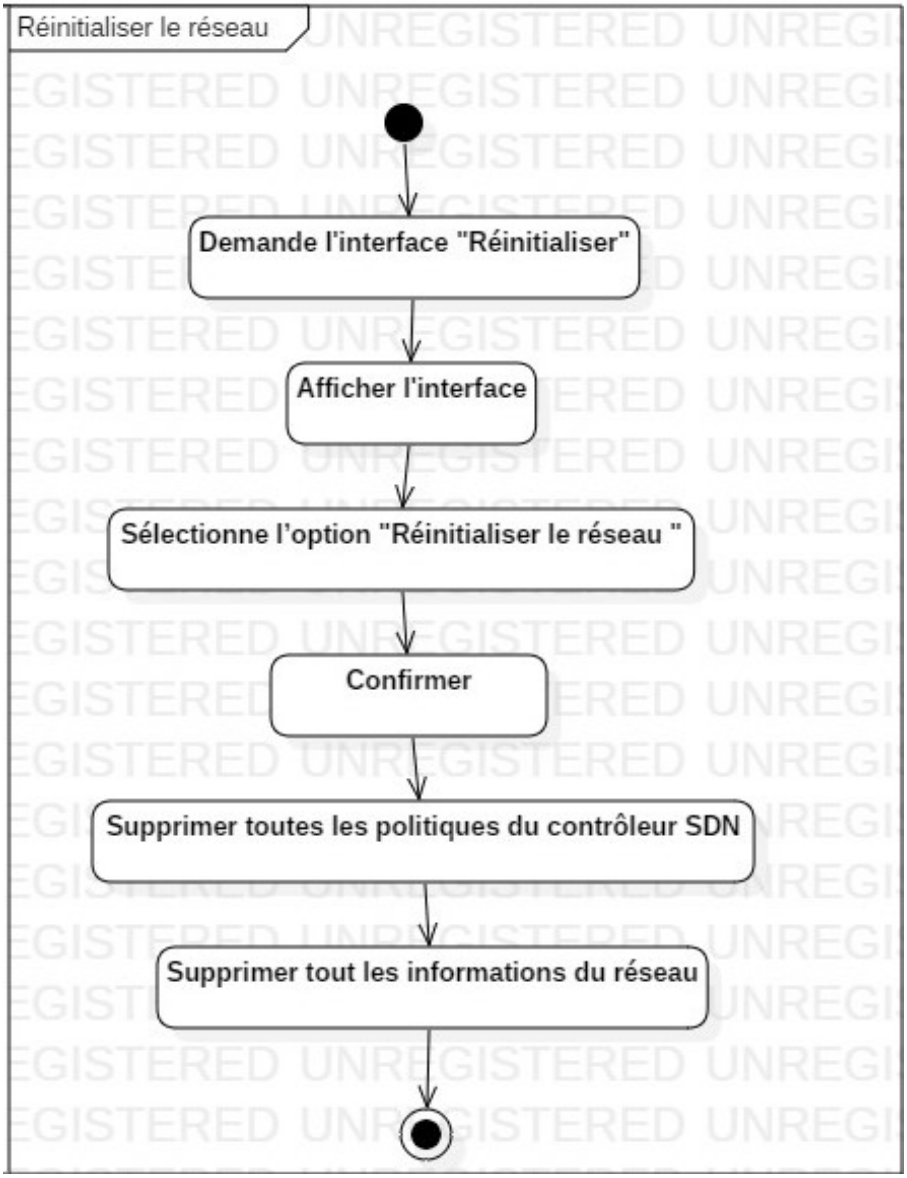

FIGURE 3.34 – Diagramme d'activité du cas d'utilisation « Réinitialiser le réseau ».

• **Cas d'utilisation « Chercher des politique » :**

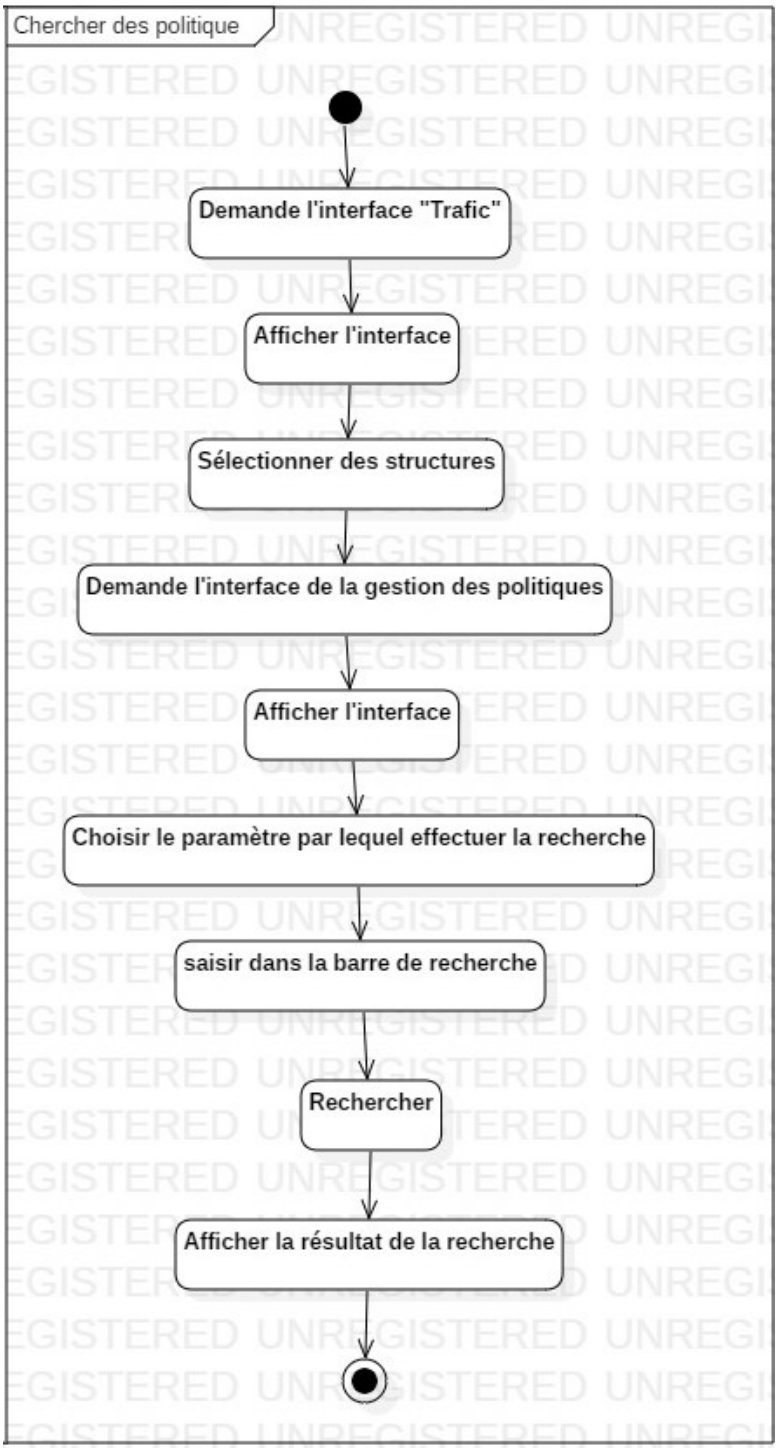

FIGURE 3.35 – Diagramme d'activité du cas d'utilisation « Chercher des politique ».

C-Phase de conception :

• Diagramme de classes :

• Définition :

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs.
En conception, le diagramme de classes représente la structure d'un code orienté objet ou,à un niveau de détail plus important, les modules du langage de développement [59].

• Les composants de base du diagramme de classe :

— **Les classes :** Sont les modules de base de la programmation orientée objet. Une classe est une représentation abstraite d'un ensemble d'objets, elle contient les informations nécessaires à la construction de l'objet. La classe peut donc être considérée comme le modèle, le moule ou la notice qui va permette la construction d'un objet.

— **L'association :** Représente une relation sémantique durable entre deux classes. Elle est modélisée par un simple trait continu, reliant les deux classes. Le fait que deux instances soient ainsi liées permet la navigation d'une instance vers l'autre, et vice versa, et peut également comporter des règles de multiplicité pour la relation.

— **Généralisations :** C'est une relation entre un élément général (super classe ou parent) et un type plus spécifique de cet élément (sous-classe ou enfant). L'implication de généraliser ce que la source hérite des caractéristiques de la cible.

— **L'agrégation :** C'est un cas particulier d'association non symétrique exprimant une relation de contenance, et représentée par une flèche en forme d'un diamant blanc pointant vers la cible ou parent classe.

— **La dépendance :** C'est une relation unidirectionnelle permettant de représenter l'existence d'un lien sémantique entre deux classes. Une classe B est en dépendance de la classe A si des éléments de la classe A sont nécessaires pour construire la classe B. La relation de dépendance se représente par une flèche ouverte pointillée.

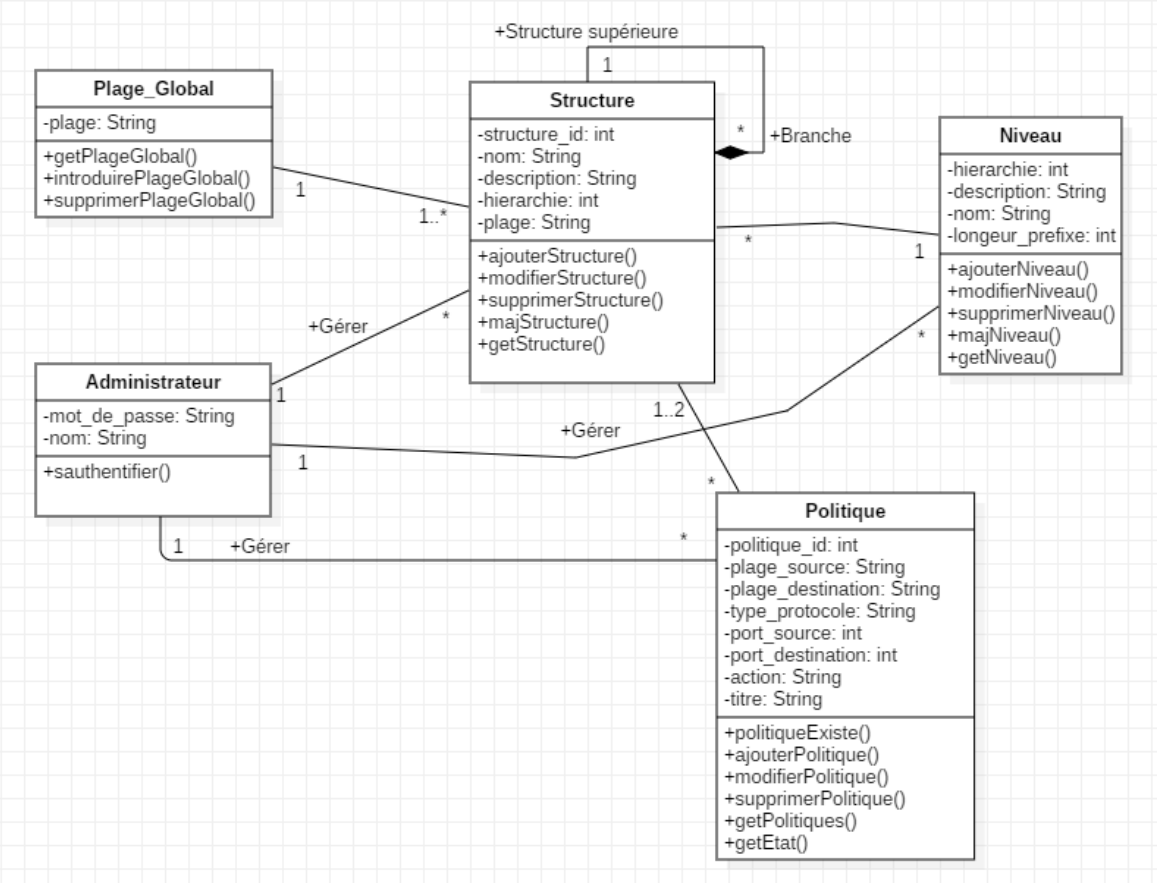

FIGURE 3.36 – Diagramme de classe du système « IPv6 Traffic-Controller ».

3.3 Implémentation et réalisation

ans cette partie nous allons présenter l'environnement logiciel, les technologies et les langages de programmation que nous avons utilisé.

3.3.1 Machine virtuelle (VirtualBox)

La VirtualBox est un package de virtualisation logicielle qui s'installe sur un système d'exploitation en tant qu'une application. La machine virtuelle permet l'installation et l'exécution des systèmes d'exploitation supplémentaires sur celui-ci, en tant que systèmes d'exploitation invités [60].

3.3.2 RYU

Ryu est un contrôleur de réseau Software defined network (SDN) ouvert conçu pour augmenter l'agilité du réseau en facilitant la gestion et l'adaptation de la gestion du trafic. Nous avons opté pour le contrôleur SDN Ryu car il présente des caractéristiques équitables, c'est le bon choix pour les applications de recherche et les petites entreprises.

3.3.3 Mininet

Pour la simulation de notre réseau SDN, nous avons utilisé mininet. Cet outil est un émulateur de réseau qui crée un réseau d'hôtes virtuels, des commutateurs, des contrôleurs et des liens. Les hôtes Mininet exécutent un logiciel réseau Linux standard et ses commutateurs prennent en charge OpenFlow pour un routage personnalisé très flexible. Il est conçu pour supporter la recherche, le développement et l'apprentissage dans les technologies SDN [61].

3.3.4 Eclipse

Eclipse est un environnement de développement intégré (EDI) libre extensible, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java, et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions [62].

3.3.5 Oracle

Oracle est un système de gestion de base de données relationnel (SGBDR), c'est un logiciel avec des caractéristiques innovantes et particulières pouvant être personnalisé en fonction des besoins. Il est fourni par Oracle Corporation et développé par Lawrence Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates [63].

3.3.6 Java EE

JEE (Java Platform, Enterprise Edition) est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles : un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java, et un ensemble d'API qui peuvent être obtenues et utilisées séparément [64].

3.4 Simulation de l'utilisation du système «IPv6 Traffic-Controller» :

Dans cette partie, nous allons présenter un guide d'utilisation, à partir des captures d'écran, pour notre solution « IPv6 Traffic-Controller ».

• La **figure 3.37** montre l'interface principale qui s'affiche après l'authentification :

FIGURE 3.37 – L'interface principale.

• La **figure 3.38** suivante montre l'interface permettant de gérer des niveaux hiérarchiques de l'entreprise, l'organisation ou autres :

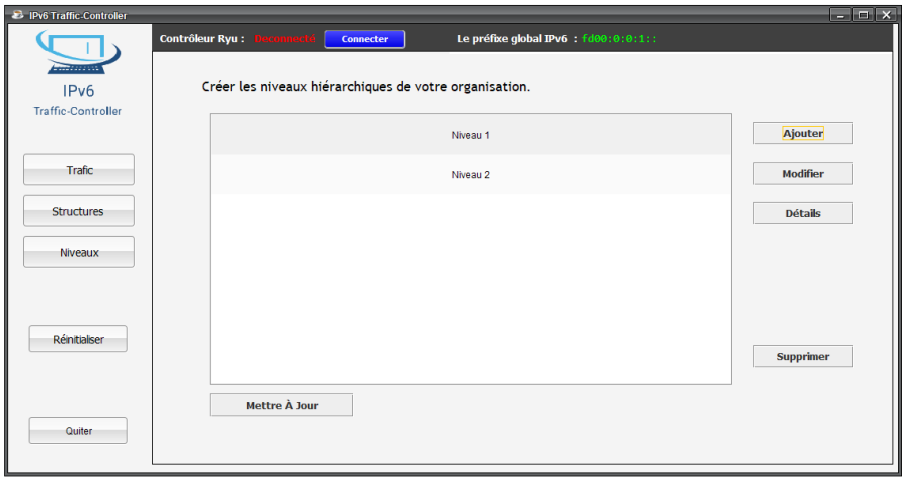

FIGURE 3.38 – l'interface « Gestion des niveaux hiérarchiques ».

• La **figure 3.39** montre la fenêtre permettant d'ajouter un niveau :

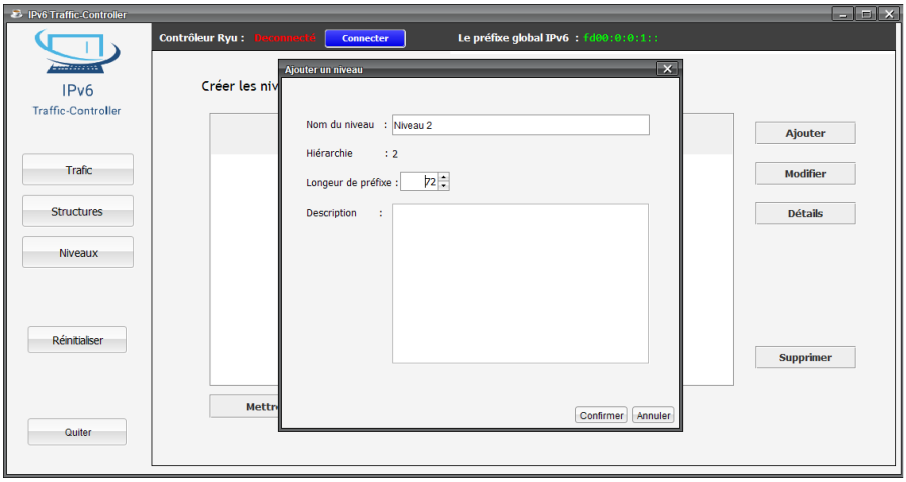

FIGURE 3.39 – La fenêtre permettant l'ajout d'un niveau.

• La **figure 3.40** montre l'interface permettant la gestion des structures organisationnelles :

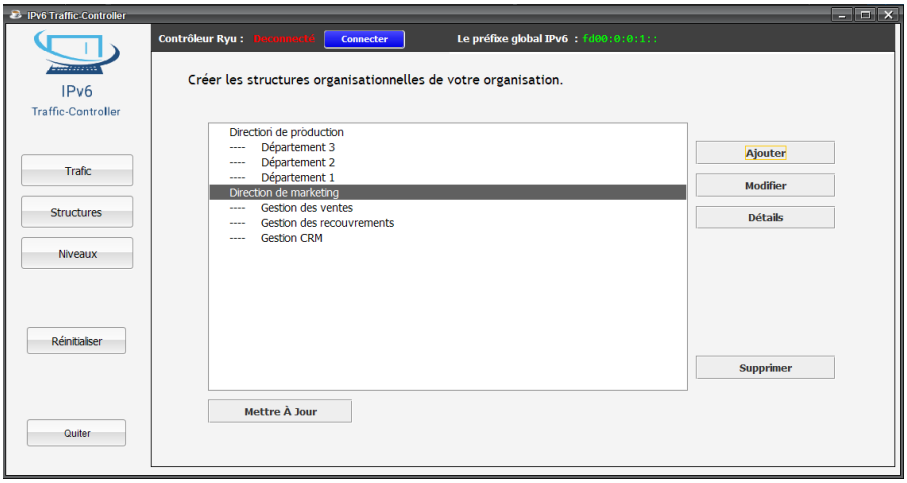

FIGURE 3.40 – L'interface permettant de gérer les structures organisationnelles.

• La **figure 3.41** montre la fenêtre permettant l'ajout d'une structure :

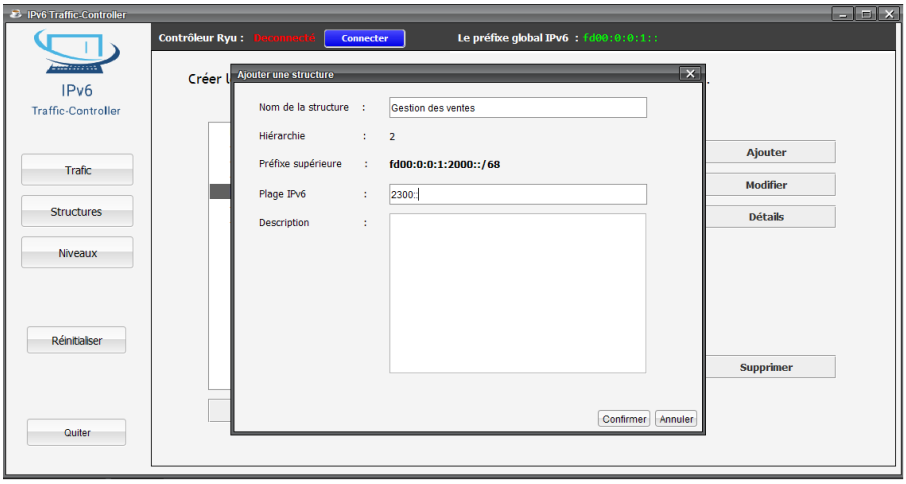

FIGURE 3.41 – La fenêtre permettant l'ajout d'une structure.

• La **figure 3.42** montre l'interface permettant de gérer le trafic IPv6. A partir de là, l'administrateur peut procéder par un des trois scénarios suivants :

— Sélectionner les structures source et destination pour gérer ou bien pour afficher les politiques reliant ces structures.

— Sélectionnez uniquement les structures source pour ajouter des politiques avec des adresses de destination personnalisées.

— Cliquer sur le bouton « Suivant » (sans sélectionner des structures) afin d'afficher toutes les politiques.

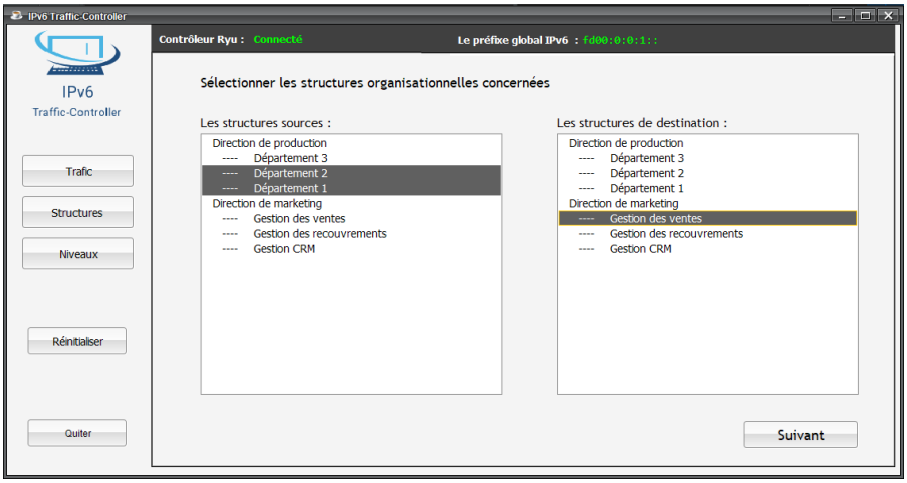

FIGURE 3.42 – L'interface permettant la gestion du trafic IPv6.

• La **figure 3.43** montre l'interface permettant la gestion des politiques :

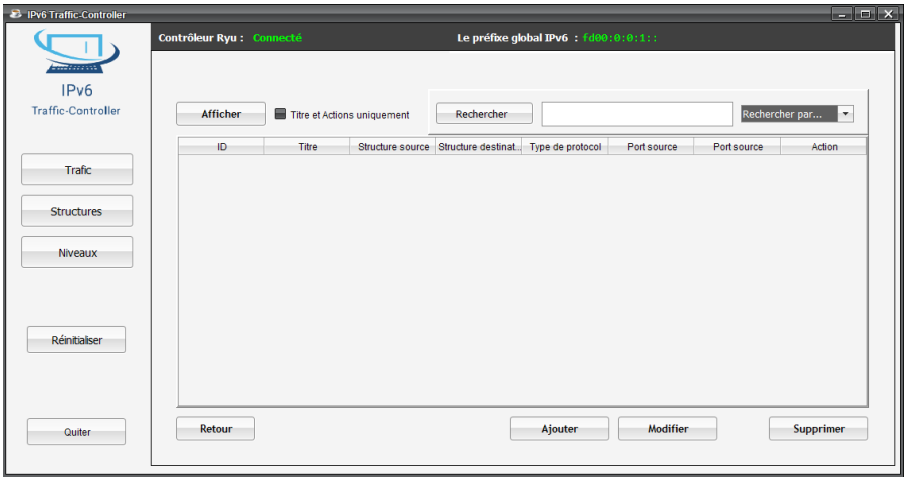

FIGURE 3.43 – L'interface permettant la gestion des politiques.

• La **figure 3.44** montre la fenêtre permettant l'ajout d'une politique :

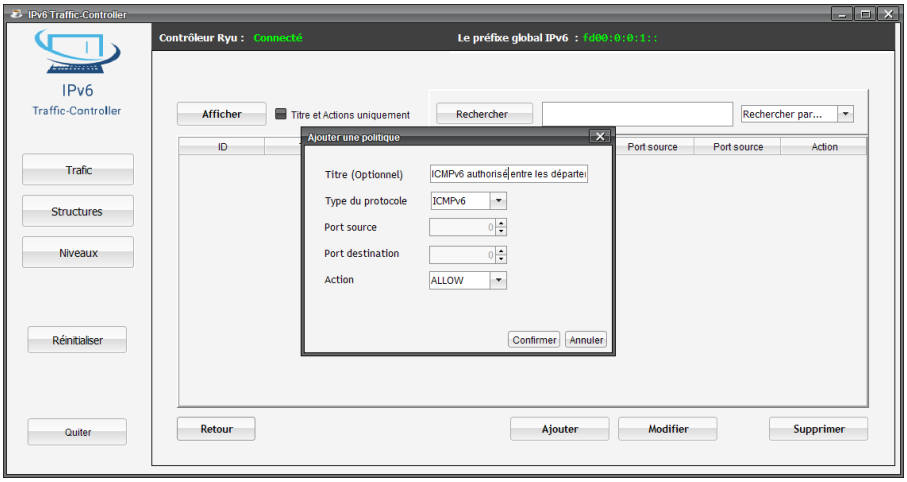

FIGURE 3.44 – La fenêtre permettant d'ajouter une politique.

• La **figure 3.45** montre l'interface affichant la liste des politiques ajoutées :

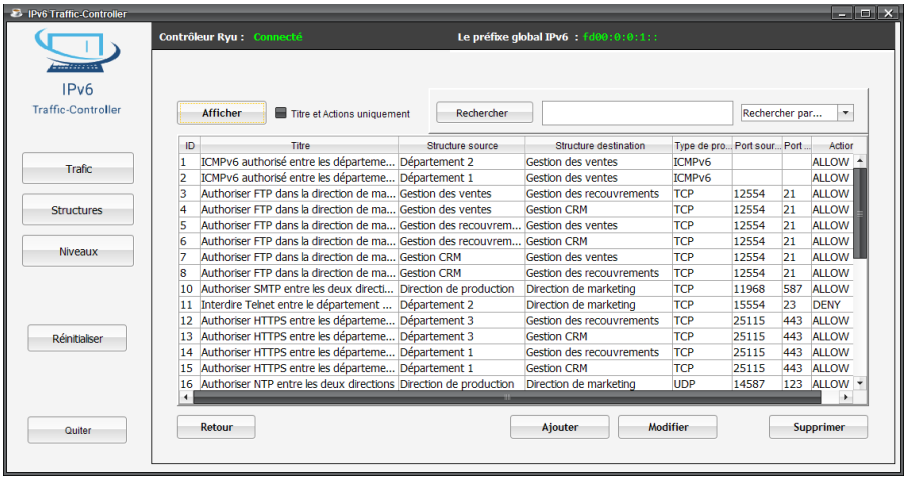

FIGURE 3.45 – Affichage des politiques.

3.5 Conclusion

Dans ce chapitre, nous avons présenté la conception détaillée de la solution « IPv6 Traffic-Controller » à travers un ensemble de diagrammes UML. Cette étape nous a permis de réaliser un projet pouvant être déployé dans un réseau SDN réel.

L'étude menée dans ce projet offre un modèle à enrichir et à étendre conduisant vers d'autres travaux dans le domaine d'automatisation des taches relatives à la gestion des réseaux.

CONCLUSION GÉNÉRALE

Dans le cadre de ce projet, nous avons développé un système permettant d'assister les administrateurs dans la tâche du contrôle de trafic IPv6 (réseau d'entreprise ou d'université) au niveau des réseaux SDN. L'approche proposée avait pour objectif d'automatiser le processus de cette tâche pour une meilleure gestion.

Notre application « IPv6 Traffic-Controller » permet une planification et une gestion hiérarchique du réseau qui reflète l'organigramme de l'organisation cible, grâce à une interface conviviale. Ce système simplifie considérablement la définition des politiques réseaux en permettant le contrôle du trafic entre les structures organisationnelles plutôt qu'entre des machines individuelles, tout en empêchant la validation des entrées contradictoires et erronées. Pour cela « IPv6 Traffic-Controller » représente une solution très bénéfique et peu coûteuse pour les opérateurs.

Dans le contexte de ce projet nous avons travaillé sur le contrôleur RYU. Donc, la partie relative à la gestion du trafic IPv6 est adaptée uniquement à ce contrôleur. Par contre, le processus de la planification du réseau est indépendant du contrôleur SDN. Nous projetons dans le futur à étendre la gestion du trafic IPv6 en développant une solution générique pouvant s'adapter aux différents contrôleurs SDN existants, et à ajouter plus de fonctionnalités concernant la gestion du plan de réseau.

Ce travail important nécessite une longue phase d'analyse, afin de bien étudier les différents APIs des applications SDN pour la gestion du trafic, aussi pour étudier le protocole de communication IPv6 ainsi que les différents protocoles réseau requis pour tous les types de données en circulation dans les organisations et les entreprises modernes. Ceci est réalisable en formant une équipe de développeurs et d'administrateurs réseaux travaillant en collaboration dans le but de contribuer à un projet de masse.

BIBLIOGRAPHIE

[1] S. Deering, R. Hinden.(2017) Internet Protocol, Version 6 (ipv6) Specification

[2] [En ligne]. Disponible : https ://www.worldipv6launch.org/apps/ipv6week/measurement/timeline-nets.html (consulté le juillet 2022)

[3] S. Tayeb,S. Latifi et Y. Kim.(2017) A Survey on IOT Communication and Computation Frameworks an Industrial Perspective

[4] B. Feldner et H. Paula.(2018) A Qualitative Evaluation of IPv6 for the Industrial Internet of Things : Semantic Scholar

[5] S. García,L. Ubiedo,T. O'Hara et M. Erguiaga.(2020)Current State of IPv6 Security in IoT

[6] [En ligne]. Disponible : https ://aws.amazon.com/vpc/ipv6/. (consulté le juillet 2022)

[7] [En ligne]. Disponible : https ://azure.microsoft.com/en-us/updates/ipv6-for-azure-virtual-network-is-now-generally-available-2/. (consulté le juillet 2022)

[8] [En ligne]. Disponible : https ://www.internetsociety.org/blog/2017/03/google-cloud-platform-gets-ipv6-support/ (consulté le juillet 2022)

[9] C. Ajay.(2021) IPv6 On Oracle Cloud Infrastructure

[10] [En ligne]. Disponible : Wireless Technology, https ://www.qualcomm.com/5g/what-is-5g/. (consulté le juillet 2022)

[11] [En ligne]. Disponible : https ://www.cio.com/article/230940/5g-connection-density-massive-iot-and-so-much-more.html. (consulté le juillet 2022)

[12] [En ligne]. Disponible : https ://www.rolandberger.com/en/Insights/Publications/Global-IPv6-and-IPv6-Development-Measurement-and-Analysis-on-Social-and.html. (consulté le juillet 2022)

[13] T. Dragos, L. Johan et V. Seppo.(2003) Routeur IPv6 TACO - une étude de cas sur la conception de processeurs de protocole

[14] K. Nichols et al.(1998) Definition of the Differentiated Services Field (DS Field) in the ipv4 and IPv6 Headers

[15] K. Ramakrishnan et al.(2001) The Addition of Explicit Congestion Notification (ECN) to IP

[16] S. Amante et al.(2011) IPv6 Flow Label Specification

[17] [En ligne]. Disponible : https ://www.ciscopress.com/articles/article.asp ?p=2803866. (consulté le juillet 2022)

[18] J.C. Mogul et J.(1985) Postel.Internet Standard Subnetting Procedure"

[19] V. Fuller.(2006) Classless Inter-domain Routing (CIDR) : The Internet Address Assignment and Aggregation Plan

[20] S. Kawamura et K. Masanobu.(2010) A Recommendation for IPv6 Address Text Representation

[21] [En ligne]. Disponible : https ://www.ibm.com/docs/en/i/7.2 ?topic=6-comparison-ipv4-ipv6. (consulté le juillet 2022)

[22] D. McPherson et al.(2014) Architectural Considerations of IP Anycast

[23] R. Hinden et S. Deering.(2006) IP Version 6 Addressing Architecture

[24] C. Huitema et B. Carpenter.(2004) Deprecating Site Local Addresses

[25] R. Hinden et B. Haberman.(2005) Unique Local IPv6 Unicast Addresses

[26] [En ligne]. Disponible : https ://study-ccna.com/what-is-ip-routing/. (consulté le juillet 2022)

[27] Z. Ashraf.(2013) IPv6 Routing : A Practitioner Approach

[28] [En ligne]. Disponible : https ://www.gdit.com/perspectives/latest/4-benefits-of-moving-to-software-defined-networking/ (consulté le juillet 2022)

[29] T. Issa et al.(2016) Etude du nomadisme dans un Cloud éducatif administré par la technologie SDN/OpenFlow 2016

[30] [En ligne]. Disponible : http ://www.efort.com (consulté le juillet 2022)

[31] [En ligne]. Disponible : https ://prosica.fr/blog/118-les-reseaux-sdn-software-defined-network.html (consulté le juillet 2022)

[32] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, et S. Uhlig.(2015) Software-defined networking : A comprehensive survey

[33] H. Djamel Eddine.(2020) Gestion de la qualité de service des flux multimédia dans les réseaux SDN

[34] B. Fouad.(2017) Etude des performances des architectures du plan de contrôle des réseaux Software-Defined Networks

[35] D. Jérome.(2015) SDN pour les nuls

[36] [En ligne]. Disponible : https ://thenewstack.io/sdn-series-part-iii-nox-the-original-openflow-controller/ (consulté le juillet 2022)

[37] [En ligne]. Disponible : https ://www.opendaylight.org (consulté le juillet 2022)

[38] [En ligne]. Disponible : https ://ryu-sdn.org/ (consulté le juillet 2022)

[39] [En ligne]. Disponible : https ://www.researchgate.net/figure/The-architecture-of-the-RYU-Controller_fig3_359024814 (consulté le juillet 2022)

[40] [En ligne]. Disponible : https ://www.businesswire.com/news/home/20200817005303/en/Global-Software-Defined-Networking-Market-2020-to-2025—Software-Defined-Networking-for-5G-Presents-Opportunities—ResearchAndMarkets.com (consulté le juillet 2022)

[41] B. Rajkumar.(2018) A Manifesto for Future Generation Cloud Computing : Research Directions for the Next Decade https ://dl.acm.org/doi/10.1145/3241737

[42] A. Aaqif Afzaal et al.(2019) Software-Defined Cloud Computing : A Systematic Review on Latest Trends and Developments

[43] [En ligne]. Disponible : https ://www.cloudflare.com/en-gb/apprentissage/couche-réseau/qu'est-ce-qu'un-wan/ (consulté le juillet 2022)

[44] Z. Yang, Yong C., Baochun L., Yadong L. et Yi X.(2020) Software-Defined Wide Area Network (SD-WAN) :Architecture, Advances and Opportunities

[45] S. Ali Haider et F. Arman Rasool.(2018) Architecture IoT assistée par SDN : un examen

[46] H. Babbar et S. Rani.(2020) Performance Evaluation of QoS metrics in Software Defined Networking using Ryu Controller https ://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012024/pdf

[47] A. Saleh, B. Goswami et S. Mohammed.(2018) Ryu Controller's Scalability Experiment on Software Defined Networks

[48] I. Tariqul, Nazrul I. et Md. Alrefat (2020) Node to node performance evaluation through RYU SDN controller

[49] M. mohammed salah et D. yaaqoub.(2019) (Route-Translator) Une solution pour le déploiement du routage IP dans les réseaux SDN

[50] [En ligne]. Disponible : http ://mininet.org/overview/ (consulte le aout 2022)

[51] Z. Houssam et B. Abdelhak.(2020) VNET_Manager : Un assistant pour la gestion des réseaux virtuels dans une infrastructure SDN

[52] [En ligne]. Disponible : https ://www.postman.com/ (consulté le août 2022)

[53] P. Roques.(2008) les Cahiers du Programmeur UML 2 Modéliser une application web

[54] [En ligne]. Disponible : https ://www.uml.org/ (consulté le août 2022)

[55] M. Laetitia.(2012) UML - Unified Modeling Language Diagrammes statiques

[56] P. Gerard.(2008) Processus de Développement Logiciel

[57] [En ligne]. Disponible : https ://sabricole.developpez.com/uml/tutoriel/unifiedProcess (consulté le août 2022)

[58] G. Joseph et G. David.(2008) UML 2 Analyse et conception : Mise en oeuvre guidée avec études de cas

[59] P. Roques.(2008) UML 2 par la pratique : étude de cas et exercices corrigés

[60] [En ligne]. Disponible : https ://www.computerhope.com/jargon/v/virtualbox.htm (consulté le août 2022)

[61] [En ligne]. Disponible : https ://www.techno-science.net/glossaire-definition/Eclipse-logiciel.html (consulté le août 2022)

[62] [En ligne]. Disponible : https ://www.techno-science.net/definition/7708.html (consulté le août 2022)

[63] [En ligne]. Disponible : https ://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm (consulté le août 2022)