الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Ref : .....

Centre Universitaire

Abd elhafid Boussouf Mila

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

Mémoire préparé en vue de l'obtention du diplôme de Master

en : Filière Informatique générale

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

Thème

Auto-Découverte Des Topologies Réseaux

Préparé par : BELHAINE Meryem

Soutenue devant le jury :

- **Président : Mr BOUKHECHEM Nadir Grade :M.A.A**
- **Examinateur : Mr DJAABOUB Salim Grade :M.A.A**
- **Encadreur : Mr BENCHEIKH EL HOCINE Madjed Grade :M.A.A**

Année universitaire : 2014/2015

DEDICACES

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

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

À celle qui m'a soutenu durant tout mes années d'étude, qui mérité mon amour éternel pour ses conseils précieuse, sa tâchasse, sa patience à ma mère *SALIHA*

À mon Père *MESSAOUD* qui a fait de moi ce que je suis, j'espère que trouveras dans modeste travail toute la fierté que peut éprouver un père pour sa fille.

À tous mes frères *KHALED, OUSSAMA, TAKIEDDINE ;*

À mon marie *SALIM ;*

À mes oncles et tantes, ainsi que mes cousins et cousins ;

À mes chère amies: *ABLA, AHLEM, FATIMA,*

Et à tous les étudiants du *STIC 2*

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

MERYEM

REMERCIEMENTS

Au terme de ce travail, je tiens vivement à remercier toutes les personnes qui ont contribué de prés ou de loin à la réalisation de ce présent travail.

En premier lieu, j'exprime toute ma gratitude pour mon promoteur, Mr ***BENCHIEKH EL HOCINE Madjed*** *pour ses précieux conseils, sa disponibilité, la confiance qu'il m'a toujours témoigné et la sollicitude dont il m'a entouré, et ce tout au long de l'élaboration du présent travail.*

Un grand merci à tous ***mes enseignants du centre universitaire Abdelhafid BOUSSOUF - Mila****, pour toutes les connaissances qu'ils m'ont inculquées.*

Sommaire

CHAPTRE III : CONCEPTION ET REALISATION

Liste des figures

Liste des tableaux

Introduction générale

INTRODUCTION GÉNÉRALE

1. Contexte

Avec le développement rapide du réseau, sa taille et sa complexité, le développement d'un système de gestion réseau devient de plus en plus nécessaire. Le système de gestion réseau est utilisé pour surveiller et contrôler le réseau entier. Il vise à assurer le bon fonctionnement et l'amélioration de l'efficacité des réseaux.

La gestion d'un réseau IP comprend la gestion des fautes, la gestion des configurations, la gestion de la comptabilité, la gestion des performances et la gestion de la sécurité. Cependant, quelque soit le domaine fonctionnel du système de gestion, l'acquisition des connaissances concernant la topologie et la connectivité entre les différents éléments du réseau joue un rôle important pour la crédibilité des résultats des systèmes de gestion réseau. En conséquence, tous les logiciels commerciaux de supervision de réseaux IP (telles HP OpenView, ou Tivoli) disposent d'un service de découverte de topologie réseau.

La découverte automatique de la topologie du réseau fait un objet d'étude très important dans ces dernières années. Elle englobe un ensemble de techniques et d'algorithmes utilisés pour la recherche automatique des périphériques du réseau (tels que les hôtes, les routeurs, les commutateurs… etc.) ainsi que la connectivité entre eux.

2. Problématique

L'administrateur réseau est confronté à des nombreuses difficultés (analyse des anomalies, détection de la source du problème et le temps nécessaire pour l'intervention) en raison de l'indisponibilité d'un outil de découverte permettant de visualiser un état actualisé sur la topologie du réseau.

3. Objectif

De nombreuses études dans le domaine de l'auto-découverte de la topologie des réseaux ont proposé des algorithmes basés sur différents protocoles tels que SNMP (Simple Network Management Protocol), ICMP (Internet Control Message Protocol) et (ARP Address Resolution Protocol)... etc.

L'objectif de ce mémoire est de développer une solution permettant de découvrir et de visualiser automatiquement la topologie (découverte du niveau trois dans le modèle OSI) d'un réseau géré par un domaine administratif privé (entreprises, organisations, ….). L'algorithme utilisé est basé sur le protocole SNMP pour l'échange des informations de gestion réseau.

Les caractéristiques devant être assurées par notre algorithme sont :

- Complet : l'algorithme doit découvrir correctement tous les réseaux accessibles.
- Cohérent : La topologie du réseau visualisée doit refléter correctement les informations collectées dans la phase de découverte.
- Efficace : l'algorithme ne doit pas consommer excessivement les ressources du réseau. Il doit être en mesure de fonctionner avec le minimum d'impact sur la bande passante du réseau.

4. Organisation du mémoire

Les chapitres de ce mémoire sont organisés comme suit :

- ✓ Le premier chapitre présente les concepts de base de la gestion réseau, permettant aux lecteurs d'avoir une vision générale sur les différents protocoles et architectures utilisés pour le développement de ces applications.

- ✓ Le deuxième chapitre présente les différentes méthodes utilisées pour la découverte automatique de la topologie réseau dans les deux niveaux : liaisons de données et réseau du modèle OSI. Ensuite présente l'algorithme de découverte proposé pour notre application.

- ✓ Le troisième chapitre présente la conception du système ainsi que l'environnement de développement utilisé pour la réalisation de notre application.

Chapitre I

La gestion des réseaux

Introduction :

Les fonctions d'un système de gestion réseau comprend la surveillance de la performance, la détection et la récupération de fautes, la configuration des ressources du réseau, le maintien de l'information comptable pour des raisons de coût et de facturation, et d'assurer la sécurité en contrôlant l'accès à la circulation de l'information dans le réseau.

Dans ce chapitre, nous allons décrire les concepts fondamentaux de la gestion des réseaux en identifiant les cinq domaines fonctionnels de base que la gestion réseau tend à couvrir. Ensuite, nous allons présenter l'architecture des systèmes de gestion basée sur le protocole SNMP.

I.1. Définition de la gestion réseau :

La gestion réseau est un service qui utilise une gamme d'outils, pour aider l'administrateur dans le contrôle et le maintien de son réseau [1]. Elle est définie comme l'exécution et le traitement d'un ensemble de fonction et d'activités requises pour la planification, le déploiement, la coordination et la surveillance des ressources réseau.

I.2. Les modèles de gestion réseau

L'organisme de normalisation internationale ISO a proposé un modèle pour l'administration réseau. Ce modèle est divisé en quatre sous-modèles comme suit [2] :

- ✓ Le modèle informationnel sert à l'identification et a la représentation des éléments à gérer (le Quoi ?).
- ✓ Le modèle organisationnel décrit les éléments participants dans la gestion (le Qui ?)
- ✓ Le modèle de communication décrit la structure des entités gérées ainsi que la façon dont les outils de gestion peuvent interagir avec ces entités (le Comment ?).
- ✓ Le modèle fonctionnel définit les différentes tâches à effectuer par la gestion (le Pourquoi ?). **(**voir la **Figure 1.1)**

Figure 1.1 : Les quatre modèles conceptuels de la gestion de réseau

I.2.1. Le modèle informationnel

Le modèle informationnel sert à l'identification et à la représentation des éléments à gérer. L'ISO a établi la structure des informations d'administration (SMI) pour définir la syntaxe et la sémantique, dont des informations de gestion stockées dans une base de donnée. Le modèle informationnel constitue la base sur laquelle s'appuient tous les autres modèles conceptuels de la gestion.

I.2.2. Le modèle organisationnel

Ce modèle décrit les composants la gestion réseau. Il base sur le concept de la relation gestionnaire/agent. Le gestionnaire et l'agent sont des processus qui échangent des informations de gestion à travers un protocole de communication. Chaque agent gère sa propre base d'informations sur laquelle le gestionnaire peut administrer. La disposition de ces composants mène à différents types d'architecture :

- **Architecture centralisée :** Chaque gestionnaire central est associé à un réseau, lequel est constitué d'un ensemble d'agents **(**voir la **Figure 1.2)**.

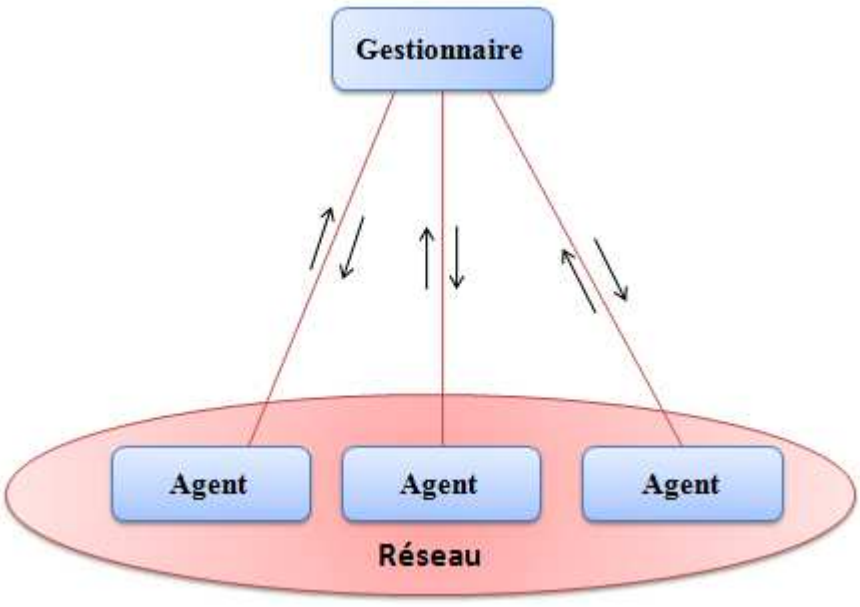

Figure 1.2 : Architecture centralisée

- **Architecture hiérarchique :** Dans cette architecture, chaque gestionnaire est responsable de la gestion de son domaine. Les gestionnaires de domaine ne communiquent pas directement entre eux, ils communiquent uniquement avec le gestionnaire central **(**voir la **Figure 1.3)**.

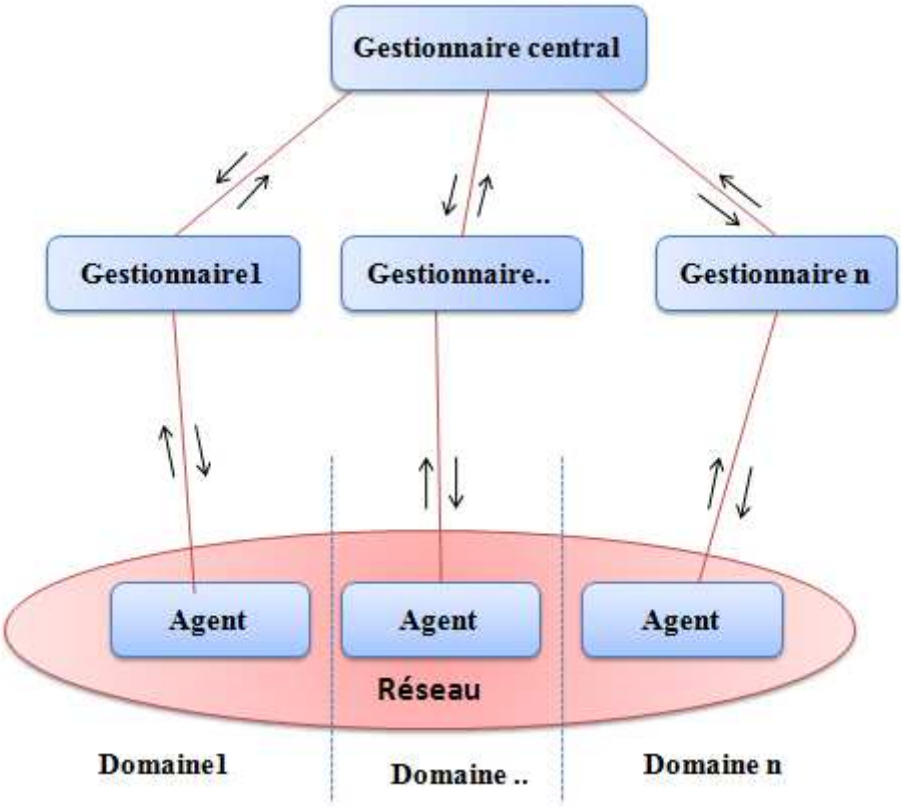

Figure 1.3 : Architecture hiérarchique

- **Architecture distribuée :** Cette architecture est constituée de plusieurs gestionnaires de domaine indépendants qui communiquent entre eux pour s'échanger de l'information sur l'état du réseau. Chacun des gestionnaires est responsable de son propre domaine **(**voir la **Figure 1.4)**.

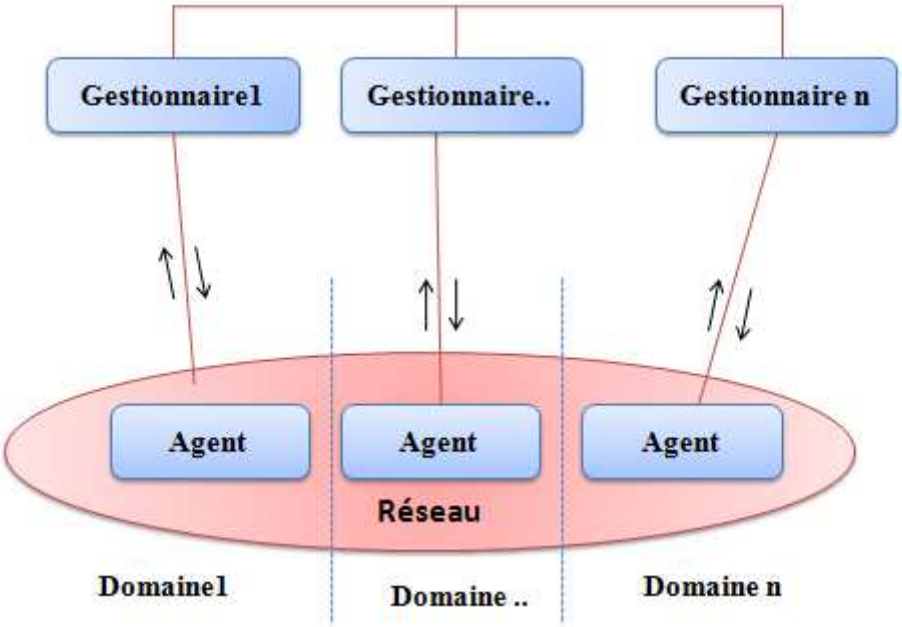

Figure 1.4 : Architecture distribuée

I.2.3. Le modèle de communication

Ce modèle traite la manière dont les données d'administration sont transmises entre les processus agent et gestionnaire. Il est relatif au protocole d'acheminement (tel que le protocole SNMP, CIMP : Common Management Information Protocol) et les messages commandes, réponses et notifications.

I.2.4. Le modèle fonctionnel

Les activités d'un système de gestion peuvent être regroupées en cinq domaines fonctionnels connus par le modèle FCAPS [1 ,3]. Pour chaque domaine, le gestionnaire réalisera la collecte des données de gestion, leur interprétation et le contrôle des éléments du réseau.

Le modèle FCAPS (**F**ault, **C**onfiguration**, A**ccounting, **P**erformance, et **S**ecurity) couvre les principales fonctions suivantes au sein de la pratique de la gestion des réseaux **(**voir la **Figure 1.5)** :

Figure 1.5 : Les cinq domaines fonctionnels d'ISO (modèle FCAPS)

I.2.4.1. La gestion des fautes (Fault Management) :

La gestion des fautes permet de localiser, isoler et réparer les pannes il y a deux façons de gérer les fautes : réactive ou proactive. Un gestionnaire réactif agit après la détection du problème. Un gestionnaire proactif assure un suivi continu de quelques paramètres critiques et veille à ce que leurs valeurs ne dépassent pas le seuil indésirable afin de réduire la possibilité d'apparition des pannes.

I.2.4.2. La gestion de la configuration (Configuration Management) :

La gestion des configurations se réfère au processus de configuration initiale d'un réseau, il permet de désigner et de paramétrer différents objets. Les procédures requises pour gérer une configuration sont la collecte d'informations, le contrôle de l'état du système et enfin la sauvegarde de l'état dans un historique.

Elle couvre l'ensemble des taches suivantes :

- ✓ Démarrage et initialisation des équipements;
- ✓ Positionnement des paramètres;
- ✓ Modification de la configuration du système.

I.2.4.3. La gestion de la comptabilité (Accounting Management) :

La gestion de la comptabilité a pour but de mesurer l'utilisation des ressources afin de réguler les accès et quantifier le taux d'utilisation des ressources. Ainsi des quotas d'utilisation peuvent être fixés temporairement ou non sur chacune des ressources réseaux. De plus, la gestion de la comptabilité autorise la mise en place de systèmes de facturation en fonction de l'utilisation pour chaque utilisateur.

I.2.4.4. La gestion des performances (Performance Management) :

La gestion des performances comprend les procédures de collecte des données et la mesure des différents aspects de la performance des équipements (taux d'erreurs, temps de transit, débit, etc.), il permet une évaluation du comportement des ressources et un contrôle de l'efficacité des activités de communication. La gestion des performances vise à garantir une meilleure qualité de service à travers une exploitation optimale des ressources.

I.2.4.5. La gestion de la sécurité (Security Management) :

La gestion de la sécurité se réfère à un ensemble de fonctions visant à protéger les réseaux et les systèmes tels que :

- ✓ Gérer les droits d'accès aux ressources
- ✓ Gestion de mécanismes de protection, de cryptage et de décryptage
- ✓ Détection de tentatives de fraude.

I.3. Système de gestion basé sur le protocole SNMP :

I.3.1. Définition de protocole SNMP :

Simple **N**etwork **M**anagement **P**rotocol est un protocole de gestion réseau IP [4]. SNMP a été introduit en 1988 pour répondre au besoin croissant d'une norme pour un protocole la de gestion des équipements réseaux, il est défini par l'Internet Engineering Task Force (IETF). Il s'agit d'un protocole de couche d'application utilisé pour la communication avec des dispositifs de réseau, pour recueillir des informations ou transmettre la configuration. SNMP est un protocole très couramment utilisé et est mis en œuvre dans la plupart des équipements de réseau disponible (par exemple, commutateurs, routeurs et serveurs…).

I.3.2. L'architecture de SNMP :

Un environnement SNMP contient généralement trois éléments de base :l'agent, la station de gestion (NMS : Network Management Station) et le MIB (Management Information Base).

I.3.2.1. Agent :

Les agents de SNMP sont des programmes installés dans les périphériques de réseau par exemple : des routeurs, des passerelles et des serveurs. Un agent SNMP est essentiellement constitué de la pile de protocoles nécessaire à la communication avec le NMS, et d'une base de données MIB.

I.3.2.2. NMS :

Le NMS désigne le périphérique utilisé par l'administrateur pour gérer son réseau. Celui-ci doit obligatoirement posséder :

• Des applications spécifiques à l'administration.

• Une interface avec l'administrateur.

La station NMS peut envoyer des requêtes à un périphérique afin d'obtenir des informations sur son paramètre. L'agent du périphérique reçoit la requête et renvoie les informations demandées. Lorsqu'elle reçoit cette réponse, la station NMS peut utiliser les informations de configuration du périphérique afin de déterminer les opérations à entreprendre en fonction de son état.

I.3.2.3. MIB :

Une MIB peut être considérée comme un magasin virtuel de l'information de gestion sous forme d'arborescence, c'est la structure qui répertorie les objets gérés. Ces objets gérés pourraient être des paramètres de configuration, de performance, et ainsi de suite.

Les relations entre ces composants sont représentées dans la figure décrite ci-dessous : **(**voir la **Figure 1.6)**.

Figure 1.6 : Architecture SNMP

I.3.3. Le modèle informationnel de SNMP

I.3.3.1. SMI:

La structure des informations de gestion définit comment chacun des éléments d'information, qui concernent les équipements gérés, est représenté dans la MIB. Les objets contenus par la MIB sont définis en utilisant le langage ASN.1 (Abstract Syntax Notation One). Chaque objet a son nom, sa syntaxe et son encodage. Le nom est l'identificateur de l'objet (Object Identifier : OID), la syntaxe définit le type de données de l'objet (ex. : entier, chaînes de caractères, tableau). Dans l'encodage l'information associée à l'objet gérée encodée selon des règles avant sa transmission sur le réseau [5,6].

I.3.3.2. Structure de MIB

La structure de la MIB est hiérarchique : les informations sont regroupées en arbre, avec des branches et des feuilles représentant les objets gérés. La MIB la plus utilisée est certainement la MIB-II, décrite dans le RFC 1213, qui définit un ensemble cohérent de paramètres communs à tous les équipements.

L'IETF utilise l'arbre d'enregistrement de l'ISO afin de nommer l'information de gestion. Chaque objet dans la MIB est identifié par un nom et un OID, L'OID est une séquence de chiffres qui parcourent l'arbre séparé par des points qui l'identifie de façon unique **(voir la figure 1.7)**.

Cet arbre est composé d'une racine à laquelle sont liés tous les nœuds.

- Le nœud-racine de l'arbre n'a pas de nom et est désigné simplement par un point [4,7].
- Le nœud ISO (1) désigne l'International Organization for Standardization.
- Le nœud ORG (3) a été créé par l'ISO à l'intention de divers organismes.
- Le nœud DOD (6) a été attribué au ministère de la Défense des États-Unis (Department Of Defense).
- Le nœud Internet (1) regroupe tout ce qui touche à l'Internet.
- Le nœud MGMT (2) fournit de différents services de normalisation.
- Le nœud MIB-2 est une sous arbre de MGMT qui contient un ensemble des nœuds subordonnés représentant différents groupements des variables MIB.

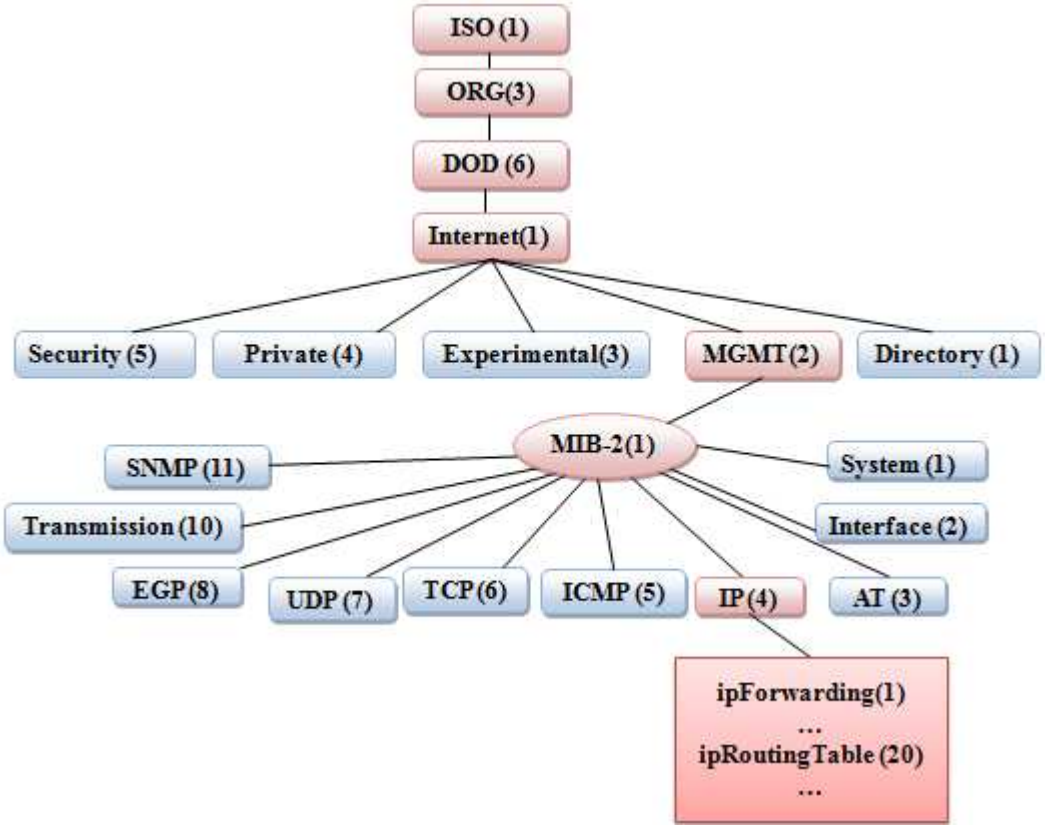

Figure 1.7 : La Structure hiérarchique de la base MIB

– Le nœud IP (4) est un groupe de MIB-2, on trouve sous ce groupe des objets, par exemple : l'objet ipRoutingTable qui indique la table de routage de l'équipement. L'OID de cette variable est obtenu en écrivant de gauche à droite les identités (nom ou bien chiffre) correspondantes aux différents nœuds, selon le format suivant :

Le nom correspondant : .iso.org.dod.internet.mgmt.mib-2.ip.ipRoutingTable →Oid= .1.3.6.1.2.1.4.20

I.3.3.3. Groupes MIB II :

Les objets gérés sont organisés en groupes [7] :

- ✓ **Le groupe système** : Fournis des informations générales sur le système géré. Par exemple, le nom, et l'emplacement du système.
- ✓ **Le groupe Interface :** Fournis les informations de configuration et les statistiques de chaque interface physique. Par exemple, le type, adresse physique, et le statut des interfaces.
- ✓ **Le groupe AT :** Traduction des adresses réseau IP en des adresses physiques MAC.
- ✓ **Le groupe IP :** Information sur la mise en œuvre et l'opération IP dans le système. Par exemple, table de routage et TTL par défaut.
- ✓ **Le groupe ICMP** : Information sur l'implémentation et l'exploitation d'ICMP. Par exemple le nombre de messages ICMP envoyés et reçus.
- ✓ **Le groupe TCP** : Information sur la l'implémentation et le fonctionnement de TCP. Par exemple, le nombre de connexions active dans le système.
- ✓ **Le groupe UDP** : Information sur la mise en œuvre et le fonctionnement de l'UDP. Par exemple, le nombre de datagrammes transmit.
- ✓ **Le groupe EGP :** Information sur la mise en œuvre et le fonctionnement d'EGP (Exterior Gateway Protocol).
- ✓ **Le groupe transmission :** Informations et statistiques sur les systèmes de transmission.
- ✓ **Le groupe SNMP** : Informations sur les accès (get, set et trap) et des erreurs d'opérations SNMP.

Dans le Tableau ci-dessous nous présentons quelque objet des différents groupes MIB-2 et leur signification [8] :

Objets MIB	Groupe MIB	OID	Signification
SysDescr	System	.1.3.6.1.2.1.1.1	Description de l'équipement
sysServices	System	.1.3.6.1.2.1.1.2	Identifier type d'équipement
ifNumber	interfaces	.1.3.6.1.2.1.2.1	nombre d'interfaces d'équipement
ipNetToMediaTable	IP	.1.3.6.1.2.1.4.22	Table ARP d'équipement.
ipDefaultTTL	IP	.1.3.6.1.2.1.4.2	Valeur utilisée dans le champ TTL
icmpInEchos	icmp	.1.3.6.1.2.1.5.8	Nbre de demandes d'écho ICMP reçues
udpInDatagrams	udp	.1.3.6.1.2.1.7.1	Nbre de datagrammes UDP reçus

Tableau 1 : Exemple d'objets MIB-II

I.3.4. Les opérations SNMP :

Les opérations de gestion réseau sont architecturées en pseudo transactions « requête/réponse » entre la station de gestion et les agents SNMP [9]. Pour manipuler l'information de gestion, les agents répondent simplement aux interrogations en renvoyant ou en changeant les données actuelles contenues dans les MIBs de leurs équipements. Le protocole SNMP possède trois types de messages : get, set et trap. Les deux premiers sont utilisés par une station de gestion pour obtenir et définir les valeurs des objets à l'agent, tandis que le dernier est utilisé par un agent pour notifier la station de gestion de certains événements **(**voir la **Figure 1.8)**.

I.3.4.1.Opération GET :

- **GetRequest (liste d'objets) :** Cette commande est émise par le gestionnaire afin de récupérer une valeur d'un objet particulier. L'agent analyse cette commande et consulte dans la MIB les objets en argument de la primitive GetRequest, ensuite l'agent répond au gestionnaire par l'envoi d'une primitive GetResponse contenant la valeur des objets demandés.

  - **GetNextRequest (liste d'objets) :** Cette commande permet de demander la valeur de l'objet suivant dans l'ordre lexicographique de l'objet passe en argument. Cette commande utilisée pour faire une lecture séquentielle des informations dans la MIB telle que la lecture des tables.

  - **GetResponse (liste d'objets et leurs valeurs) :**C'est la réponse de l'agent aux primitives GetRequest, GetNextRequest et SetRequest. Sur chaque requête du gestionnaire, l'agent répond en utilisant GetResponse. Cela peut être une réponse positive (exécutant ou confirmant l'accomplissement de l'opération demandée), ou négative dans le cas d'erreur.
  - **GetBulkRequest (SNMPv2) :** Permet de récupérer les valeurs de grands blocs d'objets. Elle a le même effet qu'une suite de message GetNextRequest, mais la bande passante utilisée est fortement réduite.

I.3.4.2.Opération SET

  - **SetRequest (liste d'objets et leurs valeurs) :** Permet de modifier des objets dans la MIB. À la réception de cette commande, l'agent met à jour les variables de la MIB à partir des valeurs en argument de SetRequest. Chacune des variables doit être précisément indiquée, et la valeur doit être en accord avec la syntaxe de la variable à modifier, sinon l'agent signale une erreur.

I.3.4.3. Opération non sollicitée

- **Trap** : C'est une commande spéciale non sollicitée, elle est émise par l'agent vers le gestionnaire pour fournir des informations concernant un événement particulier ou un état exceptionnel qui est spécifié à priori.

- **InformRequest (SNMPv2)** : Cette dernière a la particularité d'être générée d'un gestionnaire pour communiquer des informations dans son MIB en direction d'un autre gestionnaire pour permettre une gestion hiérarchique ou distribuée.

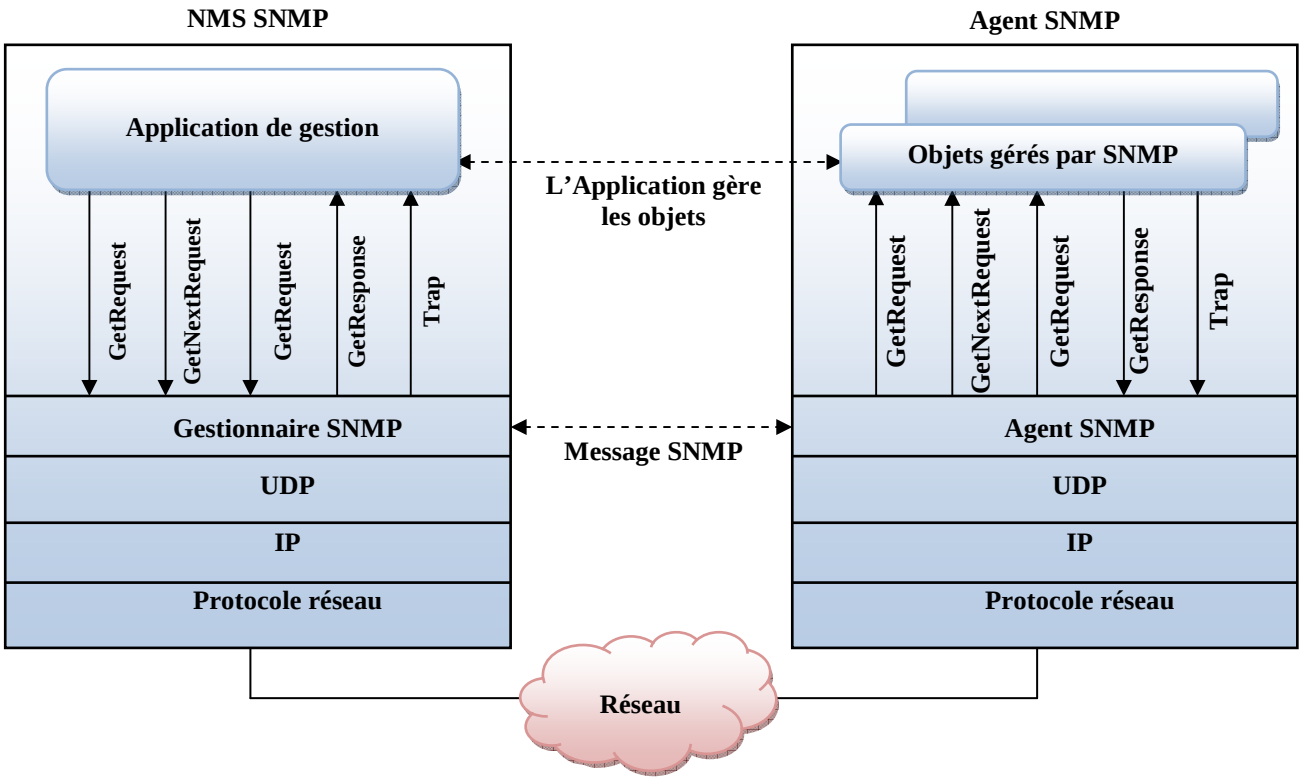

Figure 1.8 : Les opérations SNMP get, set et trap

I.3.5. SNMP et le protocole UDP :

Le protocole SNMP est basé sur UDP [4]. Chaque requête et réponse de requête représente un paquet UDP appelé message SNMP. Deux ports sont utilisés, le port 161 utilisé par l'agent pour recevoir les requêtes et envoyer les réponses de requêtes à la station de supervision, le port 162 est réservé aux messages de type trap.

1.3.6. Les versions de SNMP

Il existe trois versions de SNMP [10] :

- **SNMPv1** (décrite dans les RFC de 1155 à 1157) : C'est la première version du protocole. La sécurité de cette version est basée sur la vérification qui est faite sur la chaîne de caractères communauté (Community String). L'une des faiblesses du protocole SNMPvl est l'absence d'un mécanisme assurant la confidentialité et la sécurité des fonctions de gestion (la communauté circule en clair sur le réseau).

- **SNMPv2** (décrite dans les RFC 1901 à 1908) : Est une évolution pour améliorer la version initiale. Il offre la possibilité d'alléger la surcharge du réseau à partir de l'opération GetBulkRequest. De plus, SNMPv2 rajoute des fonctionnalités qui permettent d'obtenir des communications non seulement de gestionnaire à agent, mais aussi de gestionnaire à gestionnaire à partir l'opération informRequest.

- **SNMPv3** (décrite dans les RFC de 2571 à 2575) :C'est la dernière version, ce protocole reprend les types des opérations de la deuxième version et vise essentiellement à inclure la sécurité des transactions. Il fournit les services de sécurité tels que :

  -L'authentification :

  L'authentification a pour but d'assurer que le paquet reste inchangé pendant la transmission, et que le mot de passe de l'entité émettrice est valide [5]. Elle utilise les fonctions de hachage à une direction tel que MD5. Le NMS groupe les informations à transmettre avec un mot de passe. Ce groupe d'information passe dans la fonction de hachage à une direction. Les données obtenues et le code de hachage sont transmis sur le réseau. L'agent ajoute le mot de passe au bloc de donnée reçu. Le résultat est passé dans la fonction de hachage à une direction. Si le code de hachage est identique à celui transmis, le NMS est authentifié.

  -Le cryptage : Le cryptage a pour but d'empêcher que quelqu'un n'obtienne les informations de gestion en écoutant sur le réseau les requêtes et les réponses de quelqu'un d'autre. Avec SNMPv3, le cryptage de base se fait sur une clé partagée entre la station de supervision et l'agent. Cette clé ne doit être connue par personne d'autre.

I.3.7.Communautés SNMP (Community strings) :

L'authentification et le contrôle d'accès dans SNMP sont gérés par un mécanisme simple utilisant le Community String, c'est une chaine de caractère configurée au niveau des agents afin de garantir l'échange des informations avec les NMS [11]. Les Community Strings autorisent deux modes d'accès aux objets de la MIB (voir la **Figure 1.9)** :

- ✓ **Lecture (read/only) :** Permettant la lecture des objets de la MIB.
- ✓ **Lecture/écriture (read/write) :** Permettant la lecture et la modification des objets de la MIB.

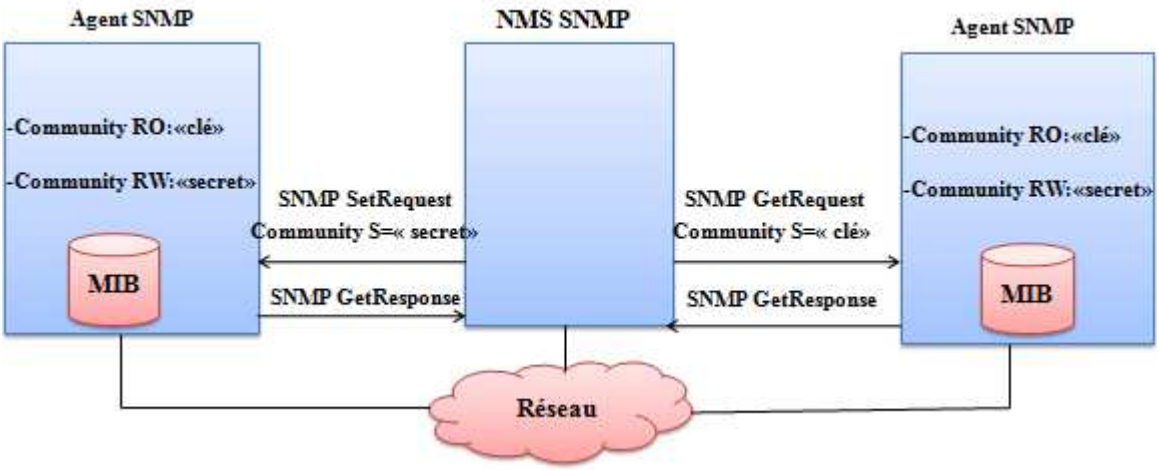

Figure 1.9 : La communication entre le NMS et les agents selon le Community strings

Conclusion

Dans la première partie de ce chapitre nous avons présenté un état de l'art sur la gestion des réseaux, à travers la description des différentes architectures, composants et protocoles tournant autour de ce domaine de recherche visant la surveillance, l'entretien et le contrôle du cycle de vie des réseaux informatiques. La deuxième partie de ce chapitre expose le protocole SNMP qui domine largement le marché des outils d'administration réseau pour sa simplicité en plus de son interopérabilité entre différentes plates-formes.

Chapitre II

La découverte de la topologie réseau dans les réseaux IP

Introduction :

La topologie du réseau est la représentation graphique de l'interconnexion entre les nœuds du réseau. Elle peut être classée en deux catégories : une topologie physique, où les nœuds sont connectés physiquement par une liaison de transmission. Et une topologie de réseau logique, qui schématise les chemins optimaux parcourus par les informations transmises.
Avoir un état mis à jour sur la topologie du réseau est une condition préalable à des nombreuses tâches de gestion, d'où le besoin d'une solution de découverte automatique est indispensable.

Il existe plusieurs techniques pour établir la découverte de la topologie réseau dans les deux catégories de découverte. Dans la première partie de ce chapitre, nous présentons un survol sur les différentes techniques de découverte utilisant des protocoles standard tels que : ICMP (Ping, Traceroute), Link Layer Discovery Protocol (LLDP), ARP et SNMP et des protocoles propriétaire tel que Cisco Discovery Protocol CDP, ensuite nous faisons une comparaison entre ces différents protocoles. Dans le deuxième partie de ce chapitre nous proposons notre algorithme qui est basé sur le protocole SNMP pour assurer la découverte de la topologie des réseaux IPv4.

II.1. La découverte automatique de la topologie réseau :

La découverte automatique de la topologie réseau est un mécanisme pour obtenir et maintenir des informations sur les nœuds existants dans le réseau (tel que les commutateurs, les routeurs, les hôtes finaux), et les informations sur l'interconnexion entre ces nœuds [12].

La découverte de la topologie réseau permet d'améliorer la planification, la gestion des ressources, la fiabilité et l'analyse de performance, ainsi que pour obtenir des avantages essentiels dans la gestion des fautes où le diagnostic de la plupart des cas d'erreur nécessite une détection automatique de la topologie du réseau.

II.2. La découverte de la topologie physique et logique :

La découverte de la topologie peut être effectuée à deux niveaux d'abstraction [13] : un niveau physique qui concerne la topologie de la couche Liaison dans le modèle OSI, et un niveau logique qui concerne la topologie de la couche réseau dans le modèle OSI :

- **La topologie du niveau physique :**
  Elle prend en compte les nœuds physiques actifs du réseau qui interviennent dans le fonctionnement et la transmission d'information au niveau de la couche Liaison, tels que les commutateurs et les terminaux

- **La topologie du niveau logique :**
  Elle représente la cartographie d'un réseau IP. Les nœuds actifs intervenant dans cette topologie sont des équipements capables d'effectuer le routage des paquets IP (par exemple : un routeur, Switch multi-layer).

D'une manière générale, la découverte de la topologie physique est principalement utilisée dans le LAN, elle établit les connexions des entités de réseau en utilisant les adresses MAC, alors que la découverte de topologie logique est principalement utilisée dans le WAN, elle établit les connexions des entités du réseau à l'aide des adresses IP.

II.3.Les techniques de découverte de la topologie

II.3.1.Les techniques active et passive

Les techniques de découverte de la topologie peuvent être classées en deux catégories [14] :

- **Une technique active :**

Elle base sur l'envoi des sondes dans le réseau ensuite l'analyse des réponses. Cette méthode influence sur la charge du réseau. Elle est caractérisée par sa faible précision

- **Une technique passive :**

Elle base sur un protocole de communication qui récupère les informations à partir des nœuds du réseau. Cette méthode consomme moins de ressources par rapport à la première méthode. Elle est caractérisée par sa grande précision.

II.3.2. Les techniques de découverte de la topologie du niveau logique

II.3.2.1. Technique basée sur SNMP

C'est une technique passive, la découverte topologie des réseaux IPv4 s'appuie sur les informations enregistrées dans les MIBs des nœuds, et plus particulièrement l'objet « ipRouteTable » de MIB II, ce objet est représenté sous forme d'un tableau bidimensionnel contenant les informations sur l'ensemble de routes installées au niveau du nœud [14].

II.3.2.2. Technique basée sur ICMP

ICMP est l'un des protocoles de base d'Internet [15]. Il est principalement utilisé pour envoyer des messages de contrôle entre des ordinateurs et des routeurs dans le réseau afin de tester l'état d'un nœud. La technique de découverte basée sur ICMP est une technique active, elle fonde sur les deux outils : le Ping et le Traceroute.

- **Le Ping :**

Son fonctionnement de base est le suivant : il envoie vers les nœuds de réseau des paquets ICMP Echo-Request, qui demande à la destination de répondre par un ICMP Echo-Reply afin de détecter les nœuds accessibles dans le réseau.

➢ Traceroute :

Cet outil utilise la spécification de la durée de vie d'un paquet IP dans le réseau (champ TTL : Time-To-Live) pour découvrir le chemin parcouru entre l'émetteur et le destinataire. Traceroute envoie une séquence des paquets ICMP Echo-Request adressée à chaque nœud détecté par le Ping, avec l'augmentation progressive de TTL, jusqu'à le dernier paquet atteint a le nœud cible, qui renvoie un ICMP Echo Reply à la source. La figure ci-dessous montre le fonctionnement de Traceroute **(**voir la **Figure 2.1)**.

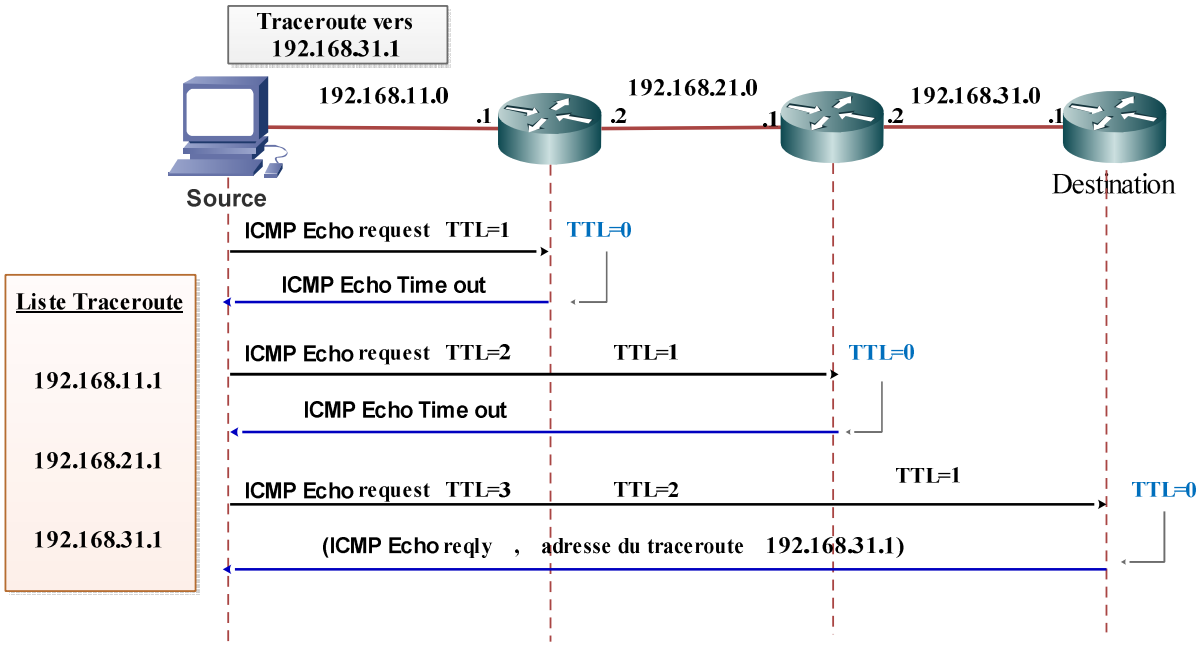

Figure 2.1 : ICMP Traceroute

.

II.3.3. Les Techniques de découverte de la topologie physique

II.3.3.1. Technique basé sur le protocole CDP et SNMP

Cisco Discovery Protocol est un protocole propriétaire développé par Cisco. L'avantage d'utiliser CDP pour découvrir la topologie est que CDP fournit des informations détaillées sur la connectivité des nœuds [16]. Chaque nœud configuré par CDP envoie périodiquement des messages, appelés annonces (avertissements), aux équipements réseau directement connectés. De plus, chaque nœud écoute les messages CDP périodiquement envoyés par leurs voisins pour déterminer leurs états.

Les informations obtenues à la fois par les voisinages ont stockée dans un Management Information Base CISCO-CDP-MIB. Chaque nœud peut recevoir des messages SNMP (Simple Network Management Protocol) pour extrait les objets CISCO-CDP-MIB **(**voir la **Figure 2.2)**.

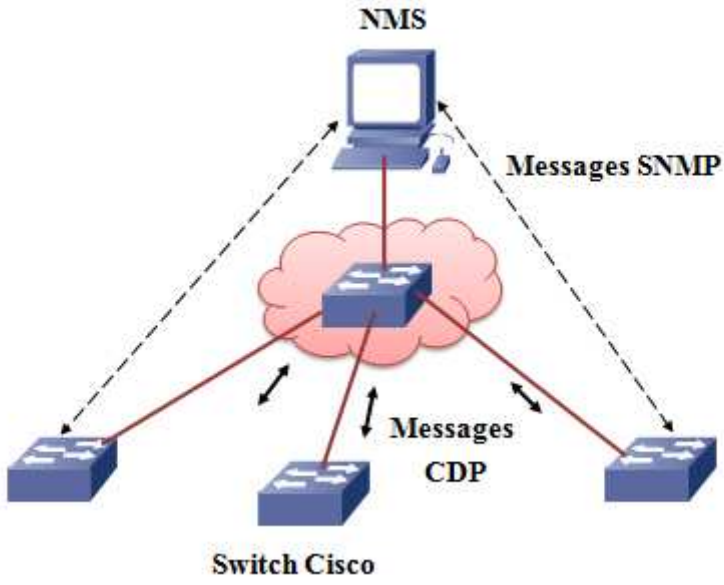

Figure 2.2: Cisco Discovery Protocol

II.3.3.2. Technique basé sur le protocole LLDP et SNMP:

Link Layer Discovery Protocol (LLDP) est un protocole de découverte Standard défini par IEEE 802.1, comme son nom l'indique il fonctionne sur la couche de liaison [17].

Ce protocole est très similaire à CDP (Cisco Discovery Protocole), Il est basé sur l'échange de trames (LLDP Data Units) spécifiques permettant à des équipements réseaux (Switch,

routeur) à annoncer et partager leurs informations avec leurs voisins. Les périphériques continuellement diffusés et écouter des messages LLDP, qu'ils puissent découvrir quand un nouveau nœud est ajouté ou retiré. De cette façon, ils maintiennent une topologie précise d'un réseau.

Les informations diffusées par ce protocole sont stockée par ses destinataires dans le MIB Standard, ce qui permet à l'information d'être accessible par un système de gestion de réseau (NMS) utilisant le protocole SNMP **(**voir la **Figure 2.3)**.

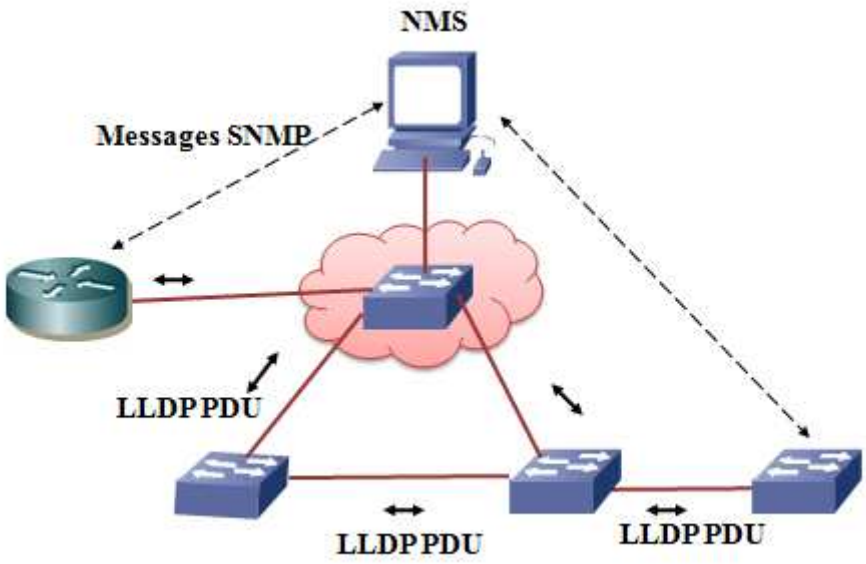

Figure 2.3 : flux de message LLDP

II.3.3.3. Technique basé sur ARP et SNMP

ARP parmi les protocoles de la couche réseau de la suite TCP/IP, a été conçu pour fournir la correspondance entre les adresses logique IP et adresses physiques MAC. Chaque nœud dans le réseau dispose un cache ARP. Par conséquent, on peut utiliser cache ARP pour découvrir touts les nœuds qui se connectent avec eux [12,14].

Le protocole SNMP utilisé l'objet "ipNetToMediaTable" dans le MIB standard. Le "ipNetToMediaTable" contient les enregistrements de cache ARP, qui interroger par le protocole SNMP pour faire découvrir les adresses IP des voisins.

II.3.4. Etude comparative des méthodes

II.3.4.1. Les critères de comparaison

Quelles que soient les techniques utilisées pour la découverte de la topologie, les critères d'évaluation de tels techniques sont toujours les mêmes. Un protocole de découverte de topologie doit remplir un certain nombre de conditions dont les principales sont [13] :

- **Précision:** Les informations de la topologie doit être correcte et en temps réel.
- **La rapidité :** Le critère de rapidité concerne Le temps d'établissement de la topologie doit être acceptable.
- **Standardisation:** La techniques doit être utilisé dans les divers du réseau.
- **Faible charge:** pour éviter d'influencer le transport des données sur le fonctionnement de réseau, la méthode de découverte de topologie de réseau doit avoir une faible charge.

II.3.4.2. Comparaison

Nous n'estimons que les techniques de découverte de topologies décrites précédemment ont des avantages et des inconvénients. En ce qui concerne les techniques du niveau logique, nous distinguons deux grandes familles : celle basée sur SNMP et celle basée sur ICMP. Le premier type de solution est généralement rapide et efficace. Le deuxième type de solution, basée sur ICMP, est très lent, ce qui entraîne par la suite une surcharge du domaine réseau à découvrir. De plus, une telle solution ne peut pas être appliquée dans un cadre réactif ou tout changement doit être signalé dans le moindre délai.

Pour les techniques de niveau physique, elles sont presque toutes basées sur l'utilisation conjointe de SNMP. Pour protocole ARP la période de mise à jour la table ARP est environ vingt minutes. Donc ce n'est pas en temps réel. L'autre est que la taille de table ARP est limite. La méthode de découverte de topologie basée sur ARP est disponible sur le LAN quand il n'existe pas plusieurs nœuds dans le réseau.

Les solutions propriétaires, malgré leurs avantages n'offrent pas un cadre générique d'application et ne sont supportées que par les équipements du constructeur (par exemple

CDP). Il reste LLDP, mais malheureusement il ne s'applique qu'au monde IEEE (Ethernet) et ne prend pas en compte d'autre technologie niveau 2 comme l'ATM.

Le Tableau ci-dessous énumère les propriétés des solutions de découverte de topologie vis-à-vis des critères que nous avons identifiés précédemment [1,4].

Critère / *méthode*	*Niveau réseau*	*Rapidité*	*précision*	*Charge réseau*	*authentification*	*Scope*	*Standardisation*
ICMP	logique	lent	milieu	élevé	aucun	milieu	standard
SNMP	Logique et physique	rapide	élevé	faible	possède	grand	standard
ARP	physique	milieu	faible	milieu	aucun	petit	standard
CDP	physique	rapide	élevé	élevé	aucun	petit	Propriétaire
LLDP	physique	rapide	élevé	élevé	aucun	petit	standard

Tableau 2.1 : Comparaison entre les différents protocoles de découverte topologie

II.4. L'algorithme proposé pour la découverte :

Dans le cadre de ce mémoire, nous proposons un système nommée **«SpiderNet»** permettant de découvrir et de visualiser automatiquement la topologie des réseaux au niveau logique.

Suite à l'étude comparative effectuée entre les différentes techniques (voir le **Tableau 2.2**), nous avons opté pour l'utilisation du protocole SNMP dans le but de réaliser une solution répondant aux besoins des administrateurs. Dans la section qui suit, nous expliquons notre algorithme qui est divisé en deux phases principales : La découverte des routeurs locaux et la découverte des routeurs distants.

II.4.1. Les objets MIB utilisés pour la découverte de la topologie

Le Tableau ci-dessous présente tous les objets SNMP (MIB-II RFC 1213) utilisés avec leurs significations :

Objet MIB De groupe IP: ipRouteTable1.3.6.1.2.1.4.21.1		
Nom de l'objet	**OID de l'objet**	**Signification**
ipRouteNextHop	1.3.6.1.2.1.4.21.1.7	L'adresse IP du prochain saut
ipRouteDest	1.3.6.1.2.1.4.21.1.1	L'adresse réseau de destination
ipRouteMask	1.3.6.1.2.1.4.21.1.11	Le masque de réseau
ipRouteIfindex	1.3.6.1.2.1.4.21.1.2	Le numéro d'interface
ipRouteProto	1.3.6.1.2.1.4.21.1.9	le protocole de routage (OSPF, RIP, …)

Tableau 2.2 : Les objets SNMP utilisés.

II.4.2. Phase 1 : La découverte automatique des routeurs locaux

A. Le but de cette phase est la détection automatique de tous les routeurs connectés au même sous-réseau que le serveur hébergeant **«SpiderNet»**. Pour réaliser cette opération, un ensemble de requêtes SNMP est envoyés à l'adresse Broadcast locale (par exemple : 192.168.21.**255** comme le montre la **Figure 2.4).**

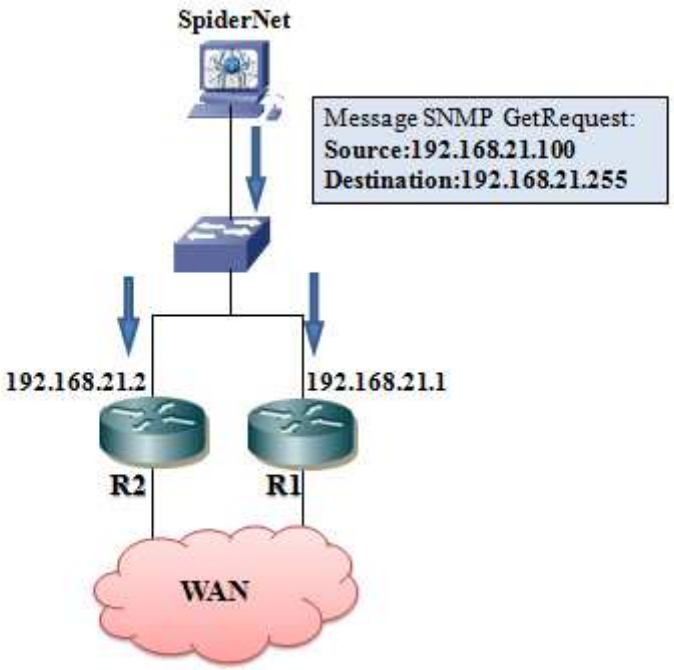

Figure 2.4 : Message broadcast SNMP GetRequest.

- Tous les routeurs connectés au même segment local reçoivent le message SNMP GetRequest, uniquement les routeurs configurés avec le bon Community String répondent avec un message SNMP GetResponse. Le système **«SpiderNet»** analyse la source des messages reçus pour déterminer les adresses IP des routeurs locaux, puis enregistrer ces adresses (voir la **Figure 2.5**).

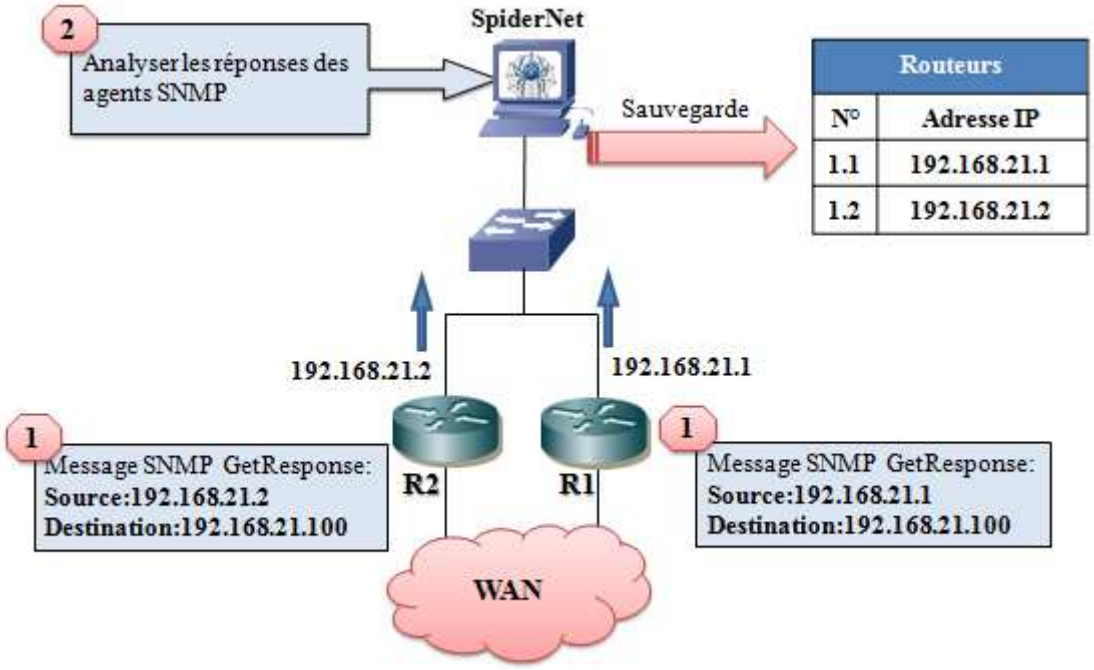

Figure 2.5 : Récupération des messages SNMP GetResponse

B. Après la détection des routeurs locaux, **«SpiderNet»** envoie des requêtes SNMP pour récupérer la table de routage de chacun des routeurs détectés **(**voir la **Figure 2.6)**.

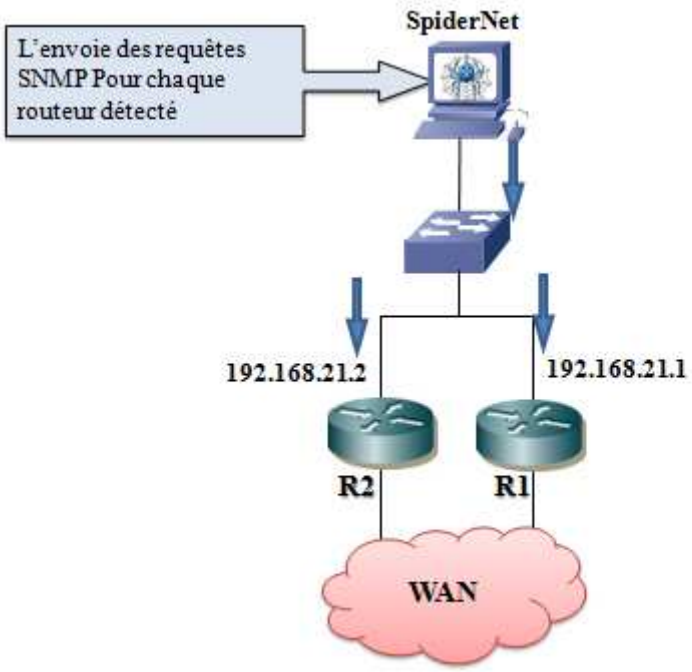

Figure 2.6: SpiderNet envoie des requêtes SNMP

- Notre système traite toutes les réponses reçu afin de sauvegarder de façon cohérente ces informations **(**voir la **Figure 2.7)**.

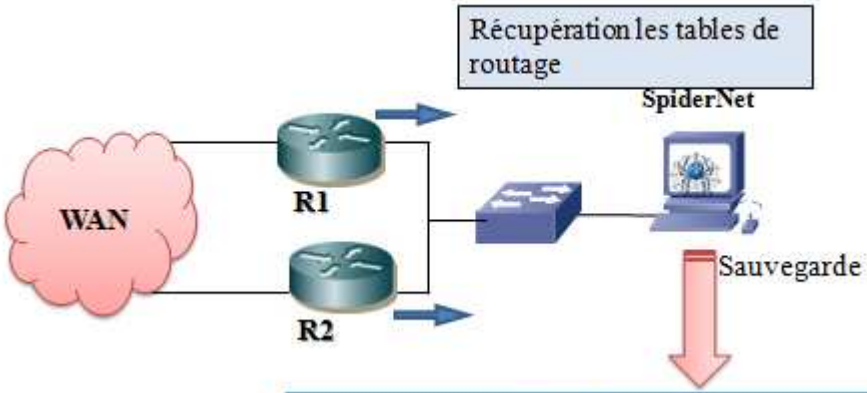

Table De Routage

N	Destination	Next-Hop	Interface	Protocole de routage
1.1	3.3.3.3	192.168.14.4	F0/0	RIP
1.1	4.4.4.4	192.168.14.4	F0/0	RIP
1.1	192.168.14.0	192.168.14.1	F1/0	directly connected
1.1	192.168.21.0	192.168. 21.1	F0/0	directly connected
1.2	5.5.5.5	192.168.25.5	F0/0	OSPF
1.2	6.6.6.6	192.168.25.5	F0/0	OSPF
1.2	192.168.25.0	192.168.25.2	F1/0	directly connected
1.2	192.168.21.0	192.168. 21.2	F0/0	directly connected

Figure 2.7: SpiderNet récupère et sauvegarde les table de routages

II.4.3. Phase 2 : Découverte des routeurs distants

Cette phase est un processus itératif très complexe. Afin de bien expliquer le déroulement de cette phase nous allons simuler le comportement du système **«SpiderNet»** avec une topologie simplifiée (voir la **Figure 2.8)** :

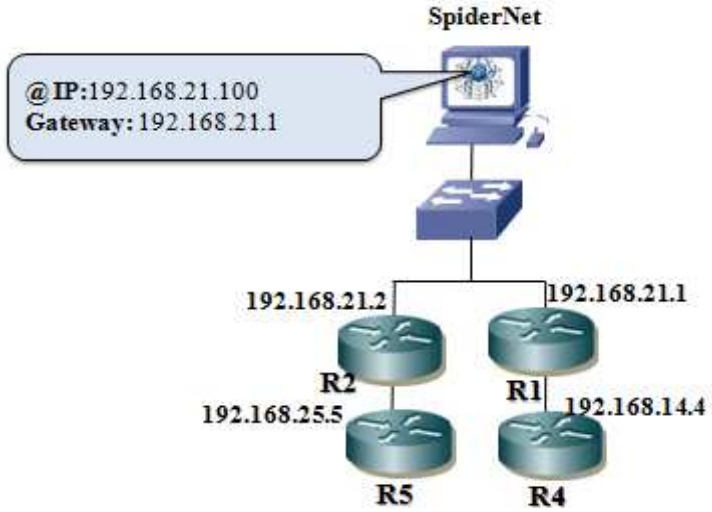

Figure 2.8: La topologie réseau

Le but de cette phase est la détection automatique de tous les routeurs se trouvant dans le réseau étendu. Pour réaliser cette opération nous utilisons l'adresse IP du prochain saut (Next-Hop) de chaque route dans la table de routage.

Comme le montre la **Figure 2.9,** la découverte des routeurs distants doit être réalisée en deux étapes (Le nombre des étapes dépond des routeurs détectés dans la phase 1) :

- **Etape N° 1** : c'est la découverte de tous les routeurs de la branche reliée avec le routeur local **R1 :192.168.21.1** →La première branche.
- **Etape N° 2**: c'est la découverte de tous les routeurs de la branche reliée avec le routeur local **R2:192.168.21.2** →La deuxième branche.

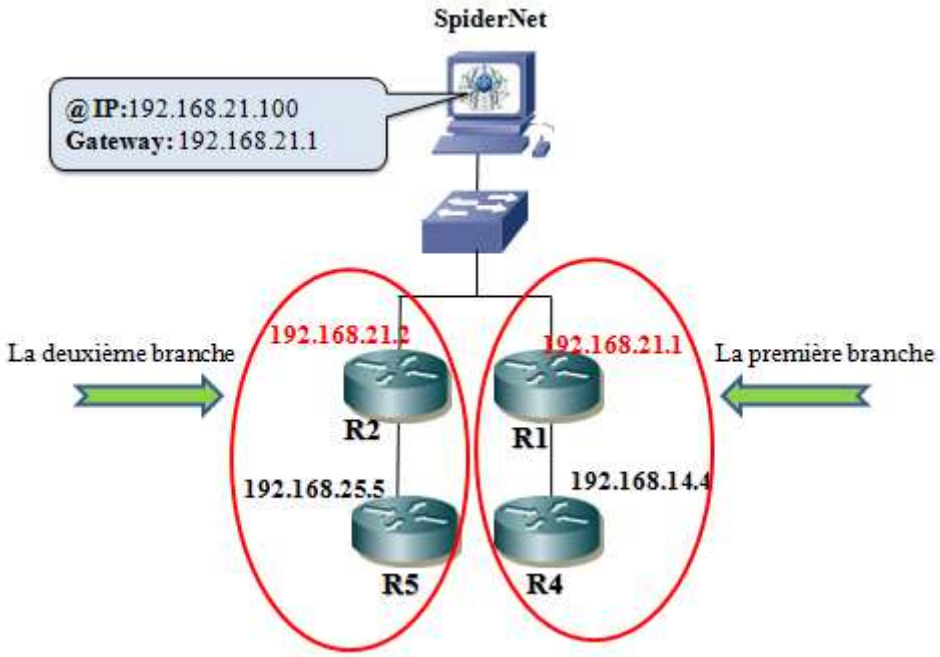

Figure 2.9 : Les branches de la topologie réseau

II.4.3.1. L'étape N° 1 : la découverte de la première branche

Pour découvrir les routeurs reliés à la première branche, le serveur hébergeant **« SpiderNet »** doit être configuré avec l'adresse IP : 192.168.21.1 de **R1** comme Gateway. A partir de la table de routage de **R1**, récupérée dans la phase 1, **« SpiderNet »** procède comme suite :

A. Un routeur peut être identifié par n'importe quelle adresse IP configurée au niveau de ses interfaces physiques ou bien logiques (interface Loopback). Ces adresses sont récupérables à partir des Next-Hop des routes directement connectées dans la table de routage. **«SpiderNet»** récupère ces adresses, ensuite il les enregistre dans la liste des routeurs visités.

B. Les Next-Hop des routes non-directement connectées dans la table de routage indiquent la liste des routeurs à découvrir. **«SpiderNet»** récupère ces adresses, ensuite il les enregistre dans la liste des routeurs non-visités (voir la **Figure 2.10**).

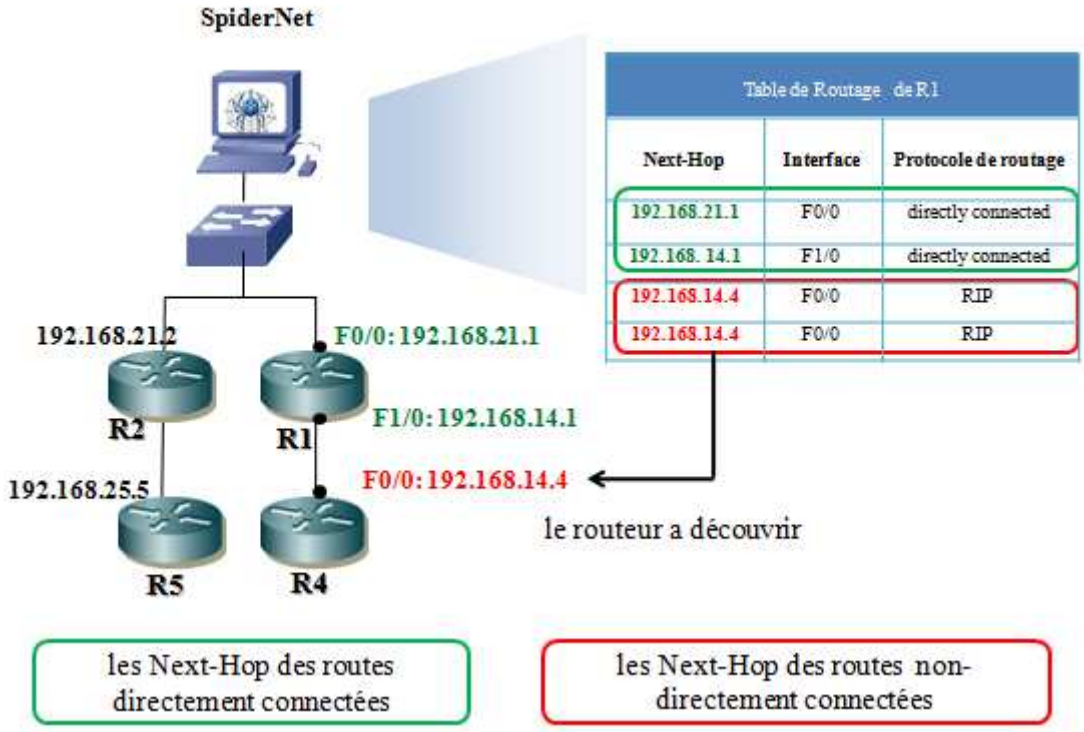

Figure 2.10 : «SpiderNet» récupère les adresses des routeurs visité et les adresses des routeurs non-visité

C. L'étape suivante consiste à récupérer la table de routage du routeur non-visité **R4 :192.168.14.4** et ainsi de suite jusqu'à la découverte de tous les routeurs connectés à la première branche. Afin de passer à la découverte de la deuxième branche nous devons obligatoirement modifier l'adresse de la Gateway du serveur hébergeant **« SpiderNet »** vers l'adresse IP : 192.168.21.1 de R2.

Le changement automatique de la Gateway du serveur hébergeant « SpiderNet » :

La configuration manuelle de la Gateway du serveur hébergeant « SpiderNet » est une opération longue, sujet à l'erreur rendant l'exploitation de notre système très lourde.

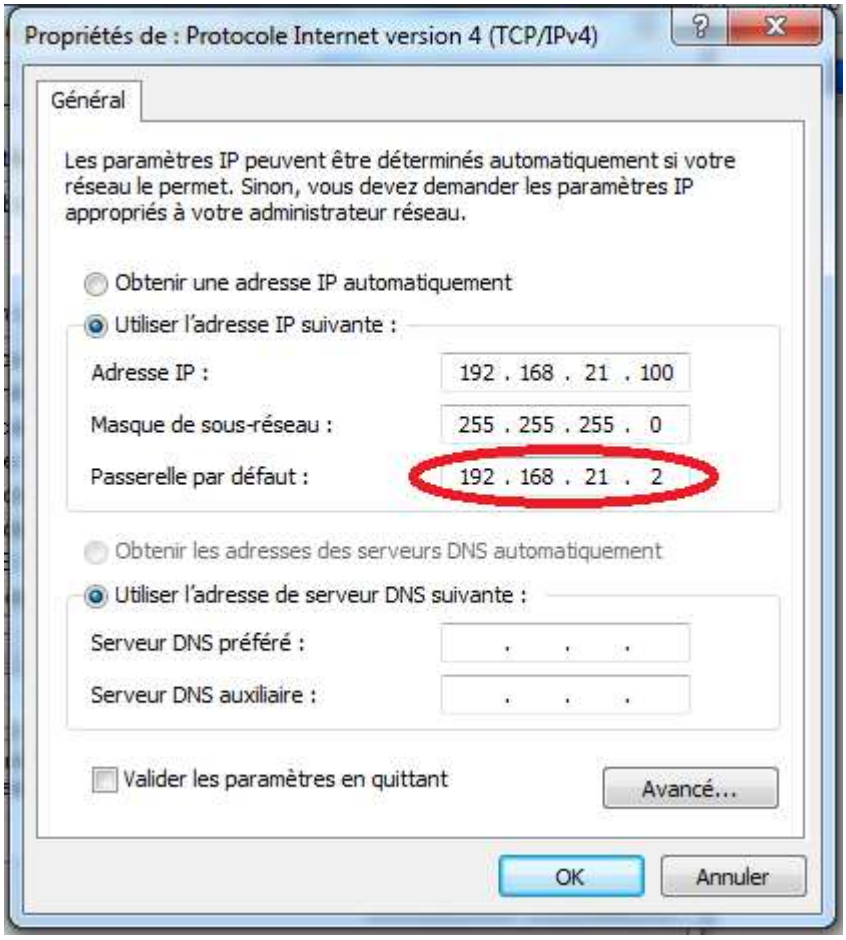

Figure 2.11 : Changement manuelle de la Gateway.

Dans le but de réaliser un système entièrement autonome, nous avons automatisé cette opération.

II.4.3.2. L'étape N° 2 : la découverte de la deuxième branche

A partir de là même processus de l'étape N°1 va réitérer jusqu'à la découverte de toutes les branches.

L'algorithme complet, proposé pour la découverte automatique des topologies est détaillé dans ce qui suit :

Algorithme de découverte

1. Auto-découverte des routeurs connectés au segment local.
2. **Si** le résultat de la phase N°1 est nul **Alors** Fin de l'algorithme, **Sinon** Ajouter les adresses IP récupérées dans la phase N°1 à la liste ***Non-Parcourue*s.**
3. Récupération des tables de routage[1] de toutes les adresses IP de la liste ***Non-Parcourue*s**.
4. A partir des tables de routages récupérées dans la phase N°3, Ajouter les Next-Hop des routes directement connectées à la liste ***Visitées.***
5. **Si** la liste ***Non-Parcourue*s** n'est pas vide **Alors** configurer la première adresse de cette liste comme Gateway au niveau du serveur hébergeant **«SpiderNet»**. **Sinon** Fin de l'algorithme.
6. A partir des tables de routages récupérées dans la phase N°3, cibler la table de routage du routeur identifié par l'adresse sélectionnée dans la phase N°5. Ajouter les Next-Hop des routes non-directement connectées qui ne figurent pas dans la liste ***Visitées*** à la liste ***Non-Visitées.***
7. **Si** la liste ***Non-Visitées*** est vide **Alors** Retirer l'adresse IP sélectionnée dans la phase N°5 de la liste ***Non-Parcourue*s**. Aller à la phase N°5. **Sinon** Récupérer la table de routage à partir du routeur identifié par la première adresse IP de la liste ***Non-Visitées.***
8. **Si** la récupération de la table de routage échoue **Alors** Ajouter le couple (adresse sélectionnée dans la phase N°7, adresse sélectionnée dans la phase N°5) à la liste ***Bloquées.*** Retirer l'adresse IP sélectionnée dans la phase N°7 de la liste ***Non-Visitées***. Aller à la phase N°7. **Sinon** Aller à la phase N°9.
9. A partir des tables de routages récupérées dans la phase N°7 :
    - Les Next-Hop des routes directement connectées sont considérées des adresses visitées. Mettre à jour les listes ***Visitées, Non-Visitées, Bloquées.***
    - Pour les Next-Hop des routes non-directement connectées, Ajouter les adresses qui ne figurent pas dans la liste ***Visitées*** à la liste ***Non-Visitées.***

    Aller à la phase N°7.

[1] L'opération de récupération de la table de routage a été exploitée à partir du système ARRC, (mémoire MASTER: Simulation d'un système de gestion de réseau autonome, HARCHE Moussa, BENYAHIA Dia-elhak, 2013/2014) avec l'autorisation de Mr BENCHEIKH El HOCINE Madjed.

Conclusion

L'auto-découverte des topologies réseaux est un outil important, efficace et pratique pour les systèmes de gestion. Dans ce chapitre nous avons présenté de façon générale les différentes techniques de découverte. Ensuite nous avons proposé un algorithme basé sur le protocole SNMP, assurant une découverte entièrement autonome des topologies réseaux au niveau trois du modèle OSI.

Chapitre III

Conception et réalisation

Introduction :

Le développement de n'importe quel projet logiciel nécessite une démarche très importante permettant de bien définir l'aspect fonctionnel des systèmes, c'est la modélisation. Il existe plusieurs langages de modélisation telle qu'UML (Unified Modeling Language), qui s'est imposé comme une norme standard dans la conception orientée objets.

Dans la première partie de ce chapitre nous allons présenter la modélisation de notre solution **«SpiderNet»** utilisant le langage UML. Ensuite, nous allons spécifier l'environnement de développement (Langages de programmation, bibliothèques, utilitaires …) qui nous a permis de réaliser ce projet.

III.1. Conception de SpiderNet :

Pour programmer une application, il ne convient pas de se lancer tête baissée dans l'écriture du code : il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. Cette démarche antérieure à l'implémentation est appelée « modélisation », son produit est un modèle.

Un modèle est une représentation abstraite et simplifiée d'un système, qui permet de mieux comprendre le système à développer. Concrètement, un modèle permet de :

- ✓ De visualiser le système comme il est ou comme il devrait l'être.
- ✓ De spécifier les structures de données et le comportement du système.
- ✓ De fournir un guide pour la construction du système.
- ✓ De documenter le système et les décisions prises.

III.1.1.Définition UML :

UML (Unified **M**odeling **L**anguage) est un langage de modélisation graphique basé sur les diagrammes. Il est apparu dans le monde du génie logiciel. Il est développé 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 [18].

Il est hérité de plusieurs méthodes telles que : OMT (Object Modeling Technique) et OOSE (Object Oriented Software Engineering) et Booch. Les principaux auteurs de la notation UML sont Grady Booch, Ivar Jacobson et Jim Rumbaugh. Il permet une couverture continue de toutes les étapes du processus logiciel **(**voir la **Figure 3.1).**

Figure 3.1 : Logo UML

III.1.2. Les vues et les diagrammes UML

UML dans sa version 2 propose treize diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel [19]. Ces types de diagrammes sont répartis en trois vues classiques **(**voir la **Figure 3.2)**.

- ✓ **Vue fonctionnelle :** permet de spécifier les tâches et les services fournis par le système.

- ✓ **Vue structurelle (statique) :** permet de visualiser, spécifier, construire et documenter l'aspect statique ou structurel du système informatisé.

- ✓ **Vue dynamique :** modélise l'aspect dynamique du système, qui montre les interactions entre le système et ses différents acteurs, ainsi que la façon dont les différents objets contenus dans le système communiquent entre eux.

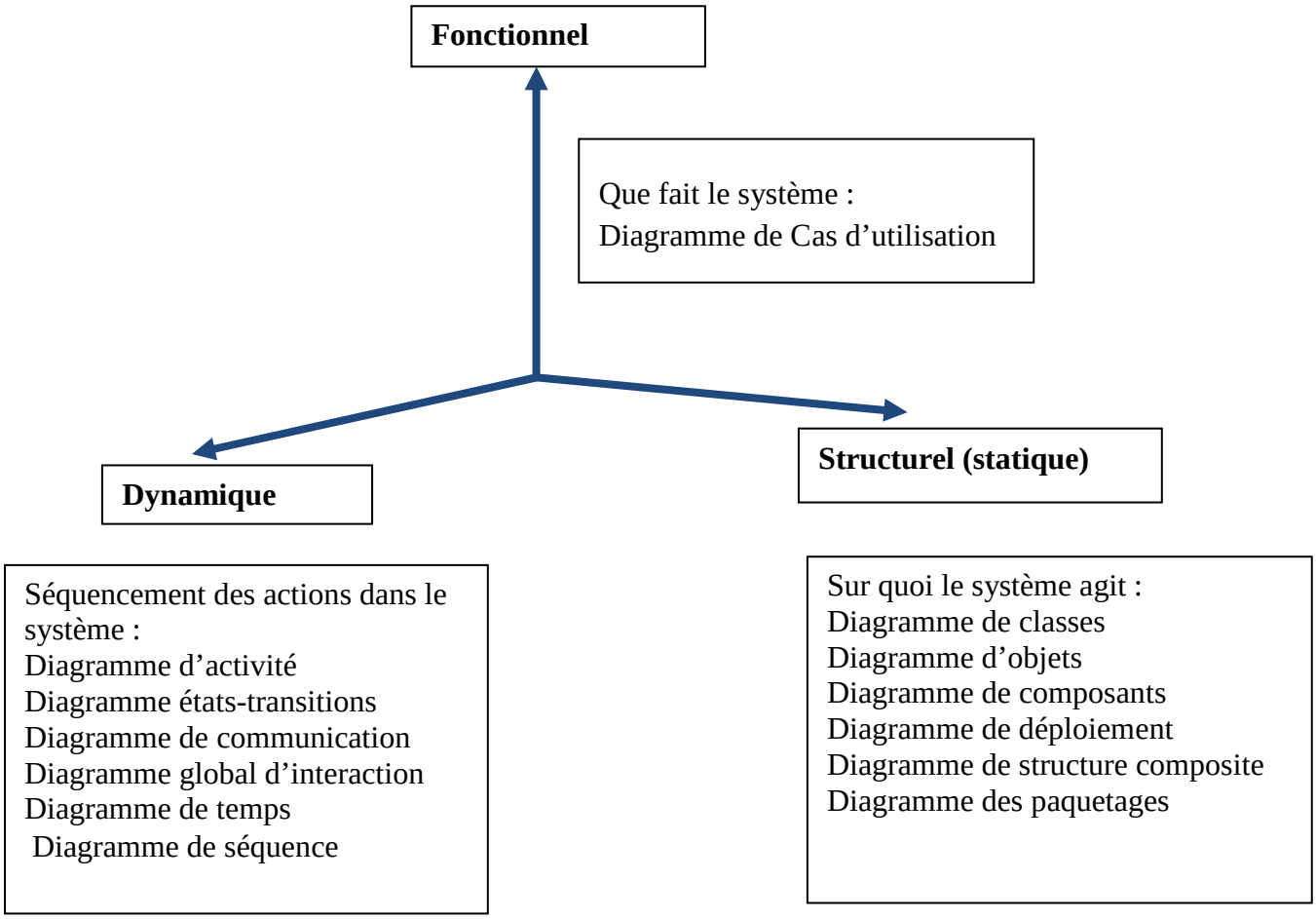

Figure 3.2 : Les trois vues classiques de modélisation

III.1.3.Processus de développement :

Un processus de développement définit une séquence d'étapes, en partie ordonnée, qui concoure à l'obtention d'un système logiciel ou à l'évolution d'un système existant, pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs.

UML n'est qu'un langage de modélisation, ce n'est pas une méthode. Pour cela, les développeurs d'UML ont créé le processus unifié (UP, Unified Process) [20].

Un processus unifié est un processus de développement logiciel basé sur UML, il est itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques.

Dans la modélisation de notre système, nous suivons une démarche simple. Cette démarche, inspirée du processus UP est répond aux exigences de notre système. Cette démarche contient trois phases :

1. Identification des besoins :

- Diagramme de cas d'utilisation

-Diagramme de séquence système

2. Phase d'analyse :

-Diagramme d'activités

3. Phase de conception :

-Diagramme global d'interaction

-Diagramme de classes

III.1.3.1.Identification des besoins :

L'objectif de cette phase est de préciser les différentes tâches du système :

➢ Diagramme de cas d'utilisation :

A- Définition :

C'est un diagramme qui identifie les grandes fonctionnalités nécessaires fournies par le système, et identifié les utilisateurs ("acteurs") du système. Le diagramme de cas d'utilisation décrit la succession des opérations réalisées par un acteur et les interactions entre les acteurs et les fonctionnalités [18].

B- Les éléments de diagramme de cas d'utilisation

-Acteur :

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. La représentation graphique standard de l'acteur en UML est l'icône appelée stick man avec le nom de l'acteur sous le dessin. Il y'a deux types d'acteur :

L'acteur principal :

- ✓ Directement concerné par le cas d'utilisation décrit
- ✓ Sollicite le système pour obtenir un résultat perceptible

Un acteur secondaire :

- ✓ Est sollicité pour des informations complémentaires
- ✓ Nécessaire au déroulement du cas d'utilisation décrit

-Cas d'utilisation

Un cas d'utilisation représente une fonctionnalité fournie par le système, modélise le service rendu par le Système sans en imposer le mode de réalisation. Les cas d'utilisation sont représentés par une ellipse contenant leur nom.

Pour détailler la dynamique du cas d'utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système. Le cas d'utilisation doit avoir un début et une fin clairement identifiés.

- Les relations :

- ➢ Relation d'association : Est un lien de communication entre un acteur et un cas d'utilisation. Elle est représentée par un trait continu.
- ➢ Relation d'inclusion : La relation d'inclusion spécifie qu'un cas d'utilisation est nécessairement une partie d'un autre cas d'utilisation. Elle est représentée par une flèche discontinue stéréotypée « inclusion ».
- ➢ Relation d'extension : La relation d'extension spécifie qu'un cas d'utilisation est éventuellement une partie d'un autre cas d'utilisation. Elle est représentée par une flèche discontinue stéréotypée « extension ».
- ➢ Relation de généralisation : La relation de généralisation/spécialisation est la transposition aux cas d'utilisation de la notion d'héritage dans le paradigme objet .Il

représenté par une flèche dont la pointe (un triangle fermé) est dirigée vers l'élément le plus général.

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

L'acteur secondaire est les routeurs de réseau qui nécessaires pour le déroulement du cas d'utilisation.

Les services offrent par**« SpiderNet »**sont résumés par les cas d'utilisation suivants :

- Authentification.
- Modifier Community String.
- Choisir L'interface.
- Auto-Découverte de la Topologie Réseau.
- Visualiser la Topologie Réseau.
- Imprimer Topologie.
- Réinitialisation.

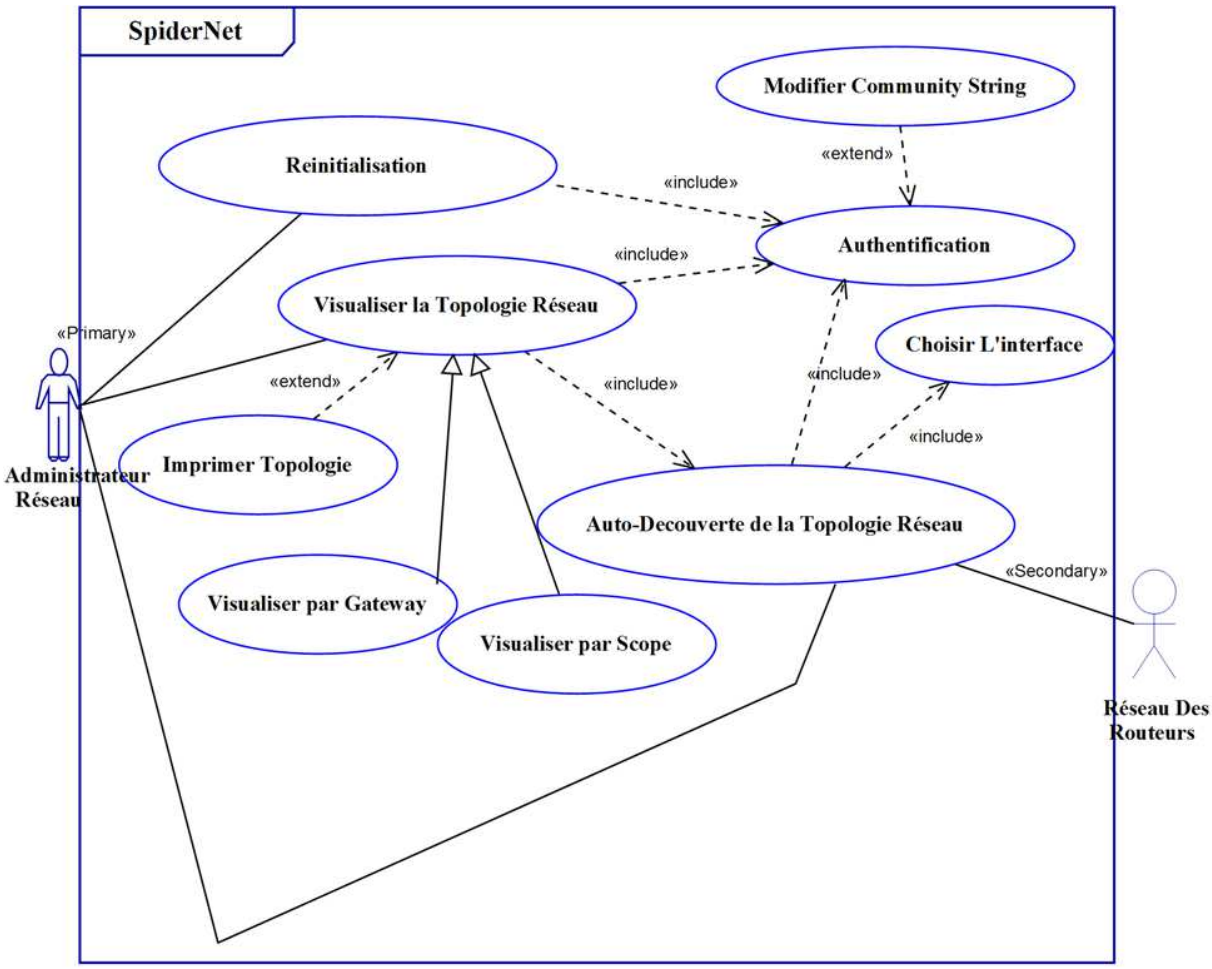

Figure 3.3 : Diagramme Cas d'utilisation du système « SpiderNet »

➢ Description textuelle d'un cas d'utilisation :

À chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et 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 articulée en six points :

• **Objectif :** Décrire 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 secondaire).

• **Pré conditions** : Si certaines conditions particulières sont requises avant l'exécution du cas, elles sont à exprimer à ce niveau.

• **Post-conditions** : Elles indiquent dans quel état se trouve le système après le déroulement de la séquence nominale

• **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 correspondants 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 (1.1, 1.2…) rappelant le numéro de l'action concernée.

❖ Cas d'utilisation « Authentification »

Acteur principal :	L'administrateur réseau.
But :	L'autorisation d'accéder au système à partir le Community String.
Pré condition :	-Le Community String existe dans la base de données. -Le Community String est configuré dans les routeurs.
Scénario nominal :	1- l'administrateur lance l'application. 2- Le système affiche le formulaire de saisir le Community String. 3- L'administrateur saisit le Community String. 4-L'administrateur valide la saisie. 5- Le système vérifie la validation le Community String. 6- Le système ouvre la page d'accueil d'application.
Scénario alternatif	5. a - Le Community String erroné. 5. b - Le système affiche un message d'erreur et retour au Scénario nominal l'étape 3.
Scénario exceptionnel :	Le système ne parvient pas à se connecter à la base de données : le système affiche un message d'erreur.

Tableau 3.1 : description textuelle de cas d'utilisation « Authentification »

❖ Cas d'utilisation « Modifier Community String »

Acteur principal :	L'administrateur réseau.
But :	Modifier le Community String.
Pré condition :	La validité de l'ancien Community String.
Scénario nominal :	1- L'administrateur lance modifier le Community String 2- Le système affiche le formulaire pour saisir les informations concernant le nouveau Community String. 3- L'administrateur remplit les deux champs : le nouveau Community String et la confirmation de nouveau Community String. 4- L'administrateur valide la modification. 5- Le système vérifie la cohérence et la validité des informations saisies et enregistre le changement de Community String.
Scénario alternatif :	5.a - Les deux champs saisis ne sont pas cohérents. 5.b- Le système affiche un message d'erreur et retour au Scénario nominal l'étape3.
Scénario exceptionnel :	Le système ne parvient pas à se connecter à la base de données : le système affiche un message d'erreur.

Tableau 3.2 : description textuelle de cas d'utilisation « Modifier Community String ».

❖ Cas d'utilisation « Choisir L'interface »

Acteur principal :	L'administrateur réseau.
But :	Choisir L'interface du serveur hébergeant **« SpiderNet»**
Pré condition :	Néant
Scénario nominal :	1- L'administrateur lance choisir l'interface. 2- Le système affiche une liste contient les noms des interfaces configurées et les informations concernant paramètres IP de chaque interface. 3- L'administrateur choisit le nom de l'interface spécifiée. 4- L'administrateur valide le choix. 4- Le système enregistre le choix.

Tableau 3.3:description textuelle de cas d'utilisation « Choisir L'interface ».

❖ Cas d'utilisation « Réinitialisation »

Acteur principal :	L'administrateur réseau.
But :	Réinitialiser la base de données.
Pré condition :	La validité du Community String.
Scénario nominal :	1- L'administrateur lance réinitialiser la base de données. 2- Le système supprime tous les enregistrements dans la BD. 3- Le système affiche une notification ("base de données est réinitialisée")
Scénario exceptionnel :	Le système ne parvient pas à se connecter à la base de données : le système affiche un message d'erreur.

Tableau 3.4 :description textuelle de cas d'utilisation « Réinitialisation ».

❖ Cas d'utilisation « Visualiser la Topologie Réseau »

Acteur principal :	L'administrateur réseau.
But :	Visualiser le schéma de la topologie réseau
Pré condition :	La validité du Community String.
Scénario nominal :	1- L'administrateur lancé visualiser la topologie. 2- Le système récupère tous les liaisons entre les routeurs et enregistrer ces liaisons. 3- Le système affiche l'écran visualisation topologie. 4- L'administrateur choisit le critère de visualisation. 3- Le système affiche la topologie réseau.
Scénario exceptionnel :	Le système ne parvient pas à se connecter à la base de données : le système affiche un message d'erreur.

Tableau 3.5: description textuelle de cas d'utilisation « Visualiser la Topologie Réseau »

❖ Cas d'utilisation « Imprimer Topologie »

Acteur principal :	L'administrateur réseau.
But :	L'impression de la Topologie .
Pré condition :	La validité du Community String.
Scénario nominal :	1-l'administrateur lance Imprimer Topologie. 2-Le système affiche la fiche d'impression.

Tableau 3.6 : description textuelle de cas d'utilisation « Imprimer Topologie ».

❖ Cas d'utilisation « Auto-Découverte la Topologie Réseau »

Acteur principal :	L'administrateur réseau.
Acteur secondaire :	Réseau des Routeurs
But :	Auto-Découverte de la Topologie Réseau
Pré condition :	-Community String valide. -l'interface est sélectionnée.
Scénario nominal :	1- L'administrateur lance l'auto-Découverte. 2- Le système passe à la phase 1 « la découverte automatique des routeurs locaux » : 2.1- Détecte les routeurs connectés au même sous-réseau puis sauvegarde les routeurs. 2.2- Récupère la table de routage de chaque routeur local puis sauvegarde cette table. 3-Le système passe à la phase 2 « La découverte des routeurs distants »: 3.1 enregistre les adresses des routeurs locaux dans la liste des routeurs visités. 3.2-La reconfiguration de Gateway du serveur hébergeant **« SpiderNet»** par chaque routeur local, et pour chaque changement de Gateway le système exécute un processus itératif jusqu'à la découverte de tous les routeurs de la branche reliée avec ce Gateway, il consiste à trois parties : **-partie1 :** Récupère les Next-Hop des routes non-directement connectées et enregistré dans la liste des routeurs non-visitées. **-Partie2 :** Récupère la table de routage du première adresse dans la liste des routeurs non-visitées. **-Partie3 :** Récupère les Next-Hop des routes directement connectées et enregistré dans la liste des routeurs visitées. Aller à la partie 1. 4- Le système affiche une notification ("la fin de découverte automatique")
Scénario alternatif :	2.a Si aucun routeur détecté, le système affiche une notification ("Liste des routeurs vide").
Scénario exceptionnel :	le système ne parvient pas à se connecter à la base de données : le système affiche un message d'erreur.

Tableau 3.7 : description textuelle de cas d'utilisation « Auto-Découverte la Topologie Réseau »

➢ Diagramme de séquence système

A-Définition

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur à savoir : saisir une donnée, consulter une donnée, lancer un traitement… etc. Il montre les interactions entre les objets selon un point de vue temporel [18,19].

Les objets communiquent en échangeant des messages représentés au moyen de flèches horizontales, orientées de l'émetteur du message vers le destinataire. L'ordre d'envoi des messages en fonction du temps est donné par la position sur l'axe vertical.

B- Les composants d'un diagramme de séquence :

Dans un diagramme des séquences, les classes et les acteurs sont énumérés en colonnes, avec leurs lignes de vie verticales indiquant la durée de vie de l'objet.

-Les objets : sont des instances des classes, et sont rangés horizontalement. La représentation graphique pour un objet est similaire à une classe (un rectangle) précédée du nom d'objet (facultatif) et deux-points (:).

-Les lignes de vie : identifient l'existence de l'objet par rapport au temps. La notation utilisée pour une ligne de vie est une ligne pointillée verticale partant de l'objet.

-Les activations : sont modélisées par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action.

-Message : modélisés par des flèches horizontales entre les activations, indiquent une communication entre des lignes de vie. Ainsi, une communication d'un objet vers un autre objet. La réception d'un message est considérée par l'objet récepteur comme un événement qu'il faut traiter (ou pas). Plusieurs types de messages existent, les plus communs sont : message synchrone, et message asynchrone

❖ Cas d'utilisation « Authentification »

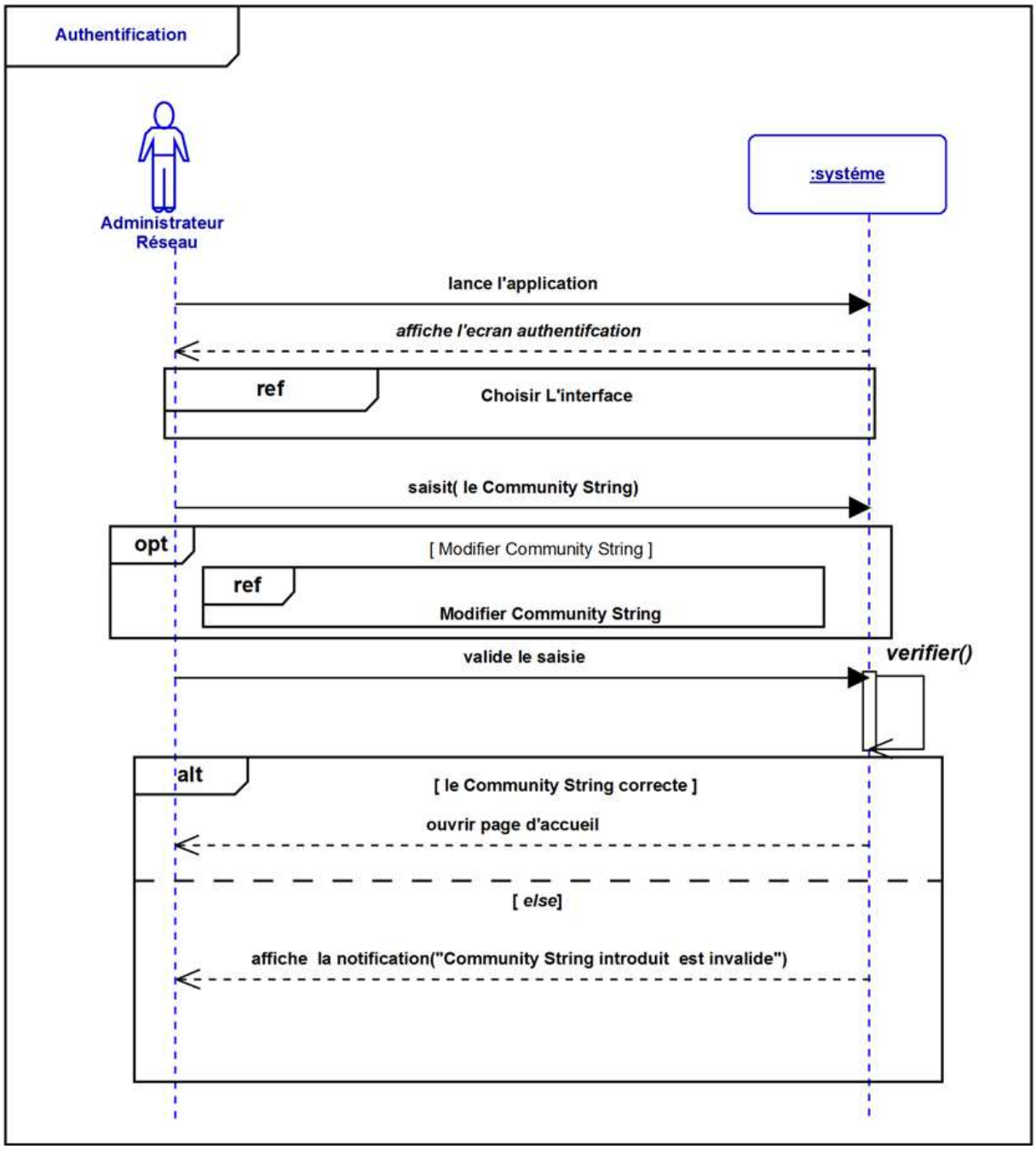

Figure 3.4 : Diagramme de séquence de Cas d'utilisation « Authentification »

❖ Cas d'utilisation « Modifier Community String »

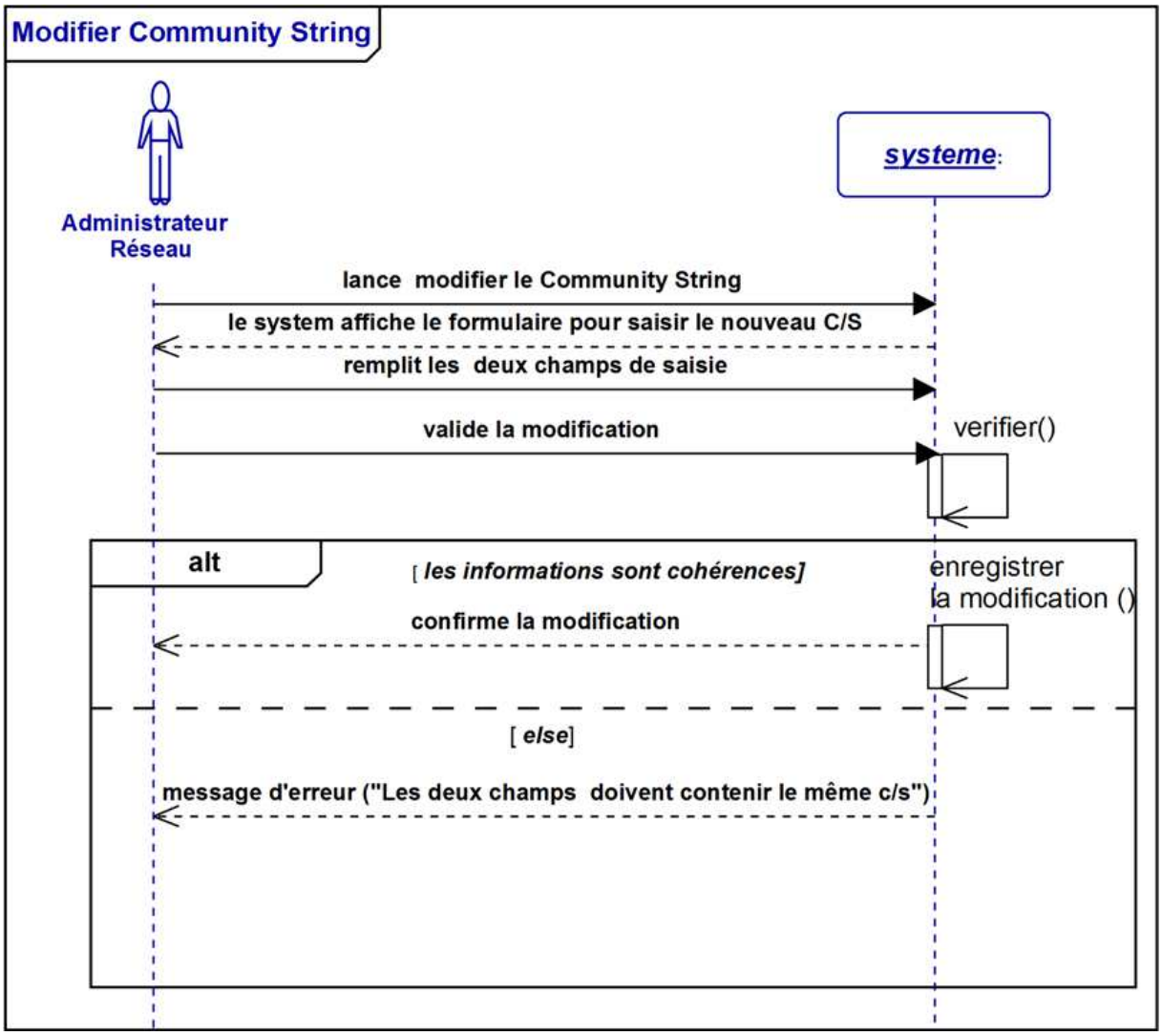

Figure 3.5 : Diagramme de séquence de Cas « Modifier Community String »

- **Cas d'utilisation « Choisir L'interface »**

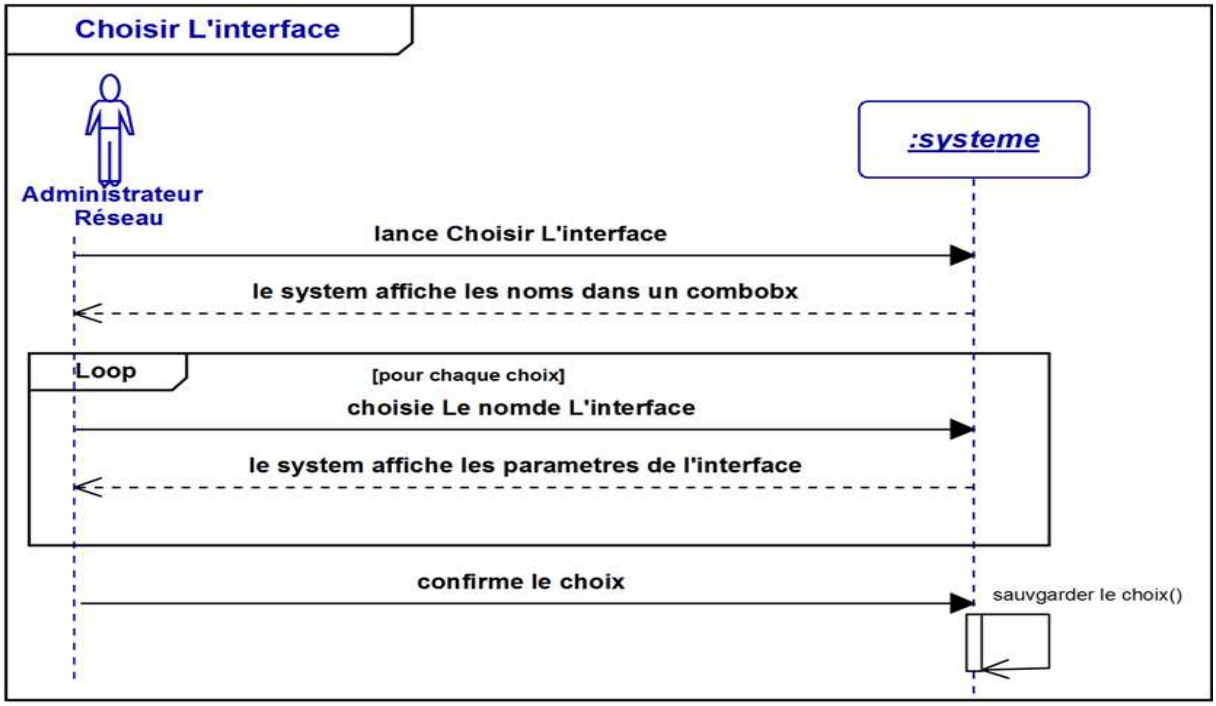

Figure 3.6 : Diagramme de séquence de Cas « Choisir L'interface »

- **Cas d'utilisation « Réinitialisation »**

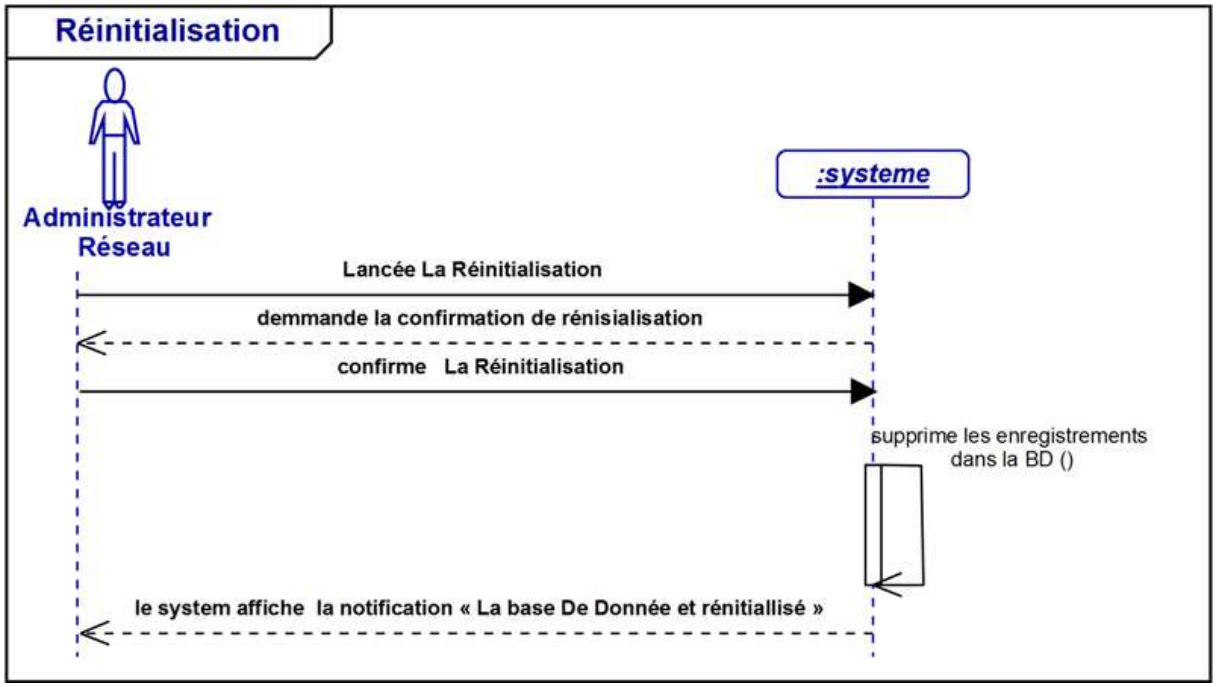

Figure 3.7 : Diagramme de séquence de Cas « Réinitialisation »

- **Cas d'utilisation « Auto-Découverte la Topologie Réseau »**

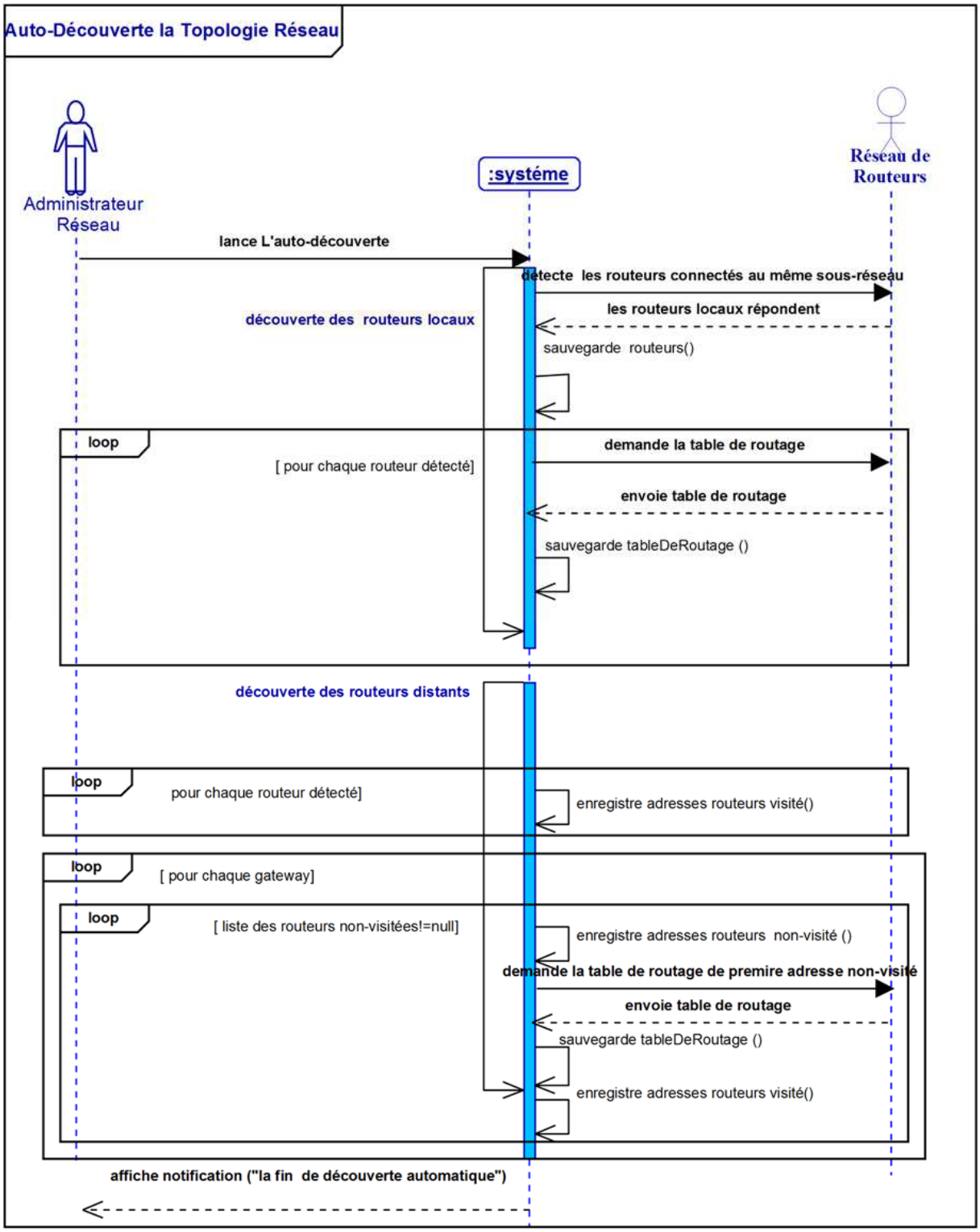

Figure 3.8 : Diagramme de séquence Cas « Auto-Découverte la Topologie Réseau »

❖ **Cas d'utilisation « Visualiser la Topologie Réseau »**

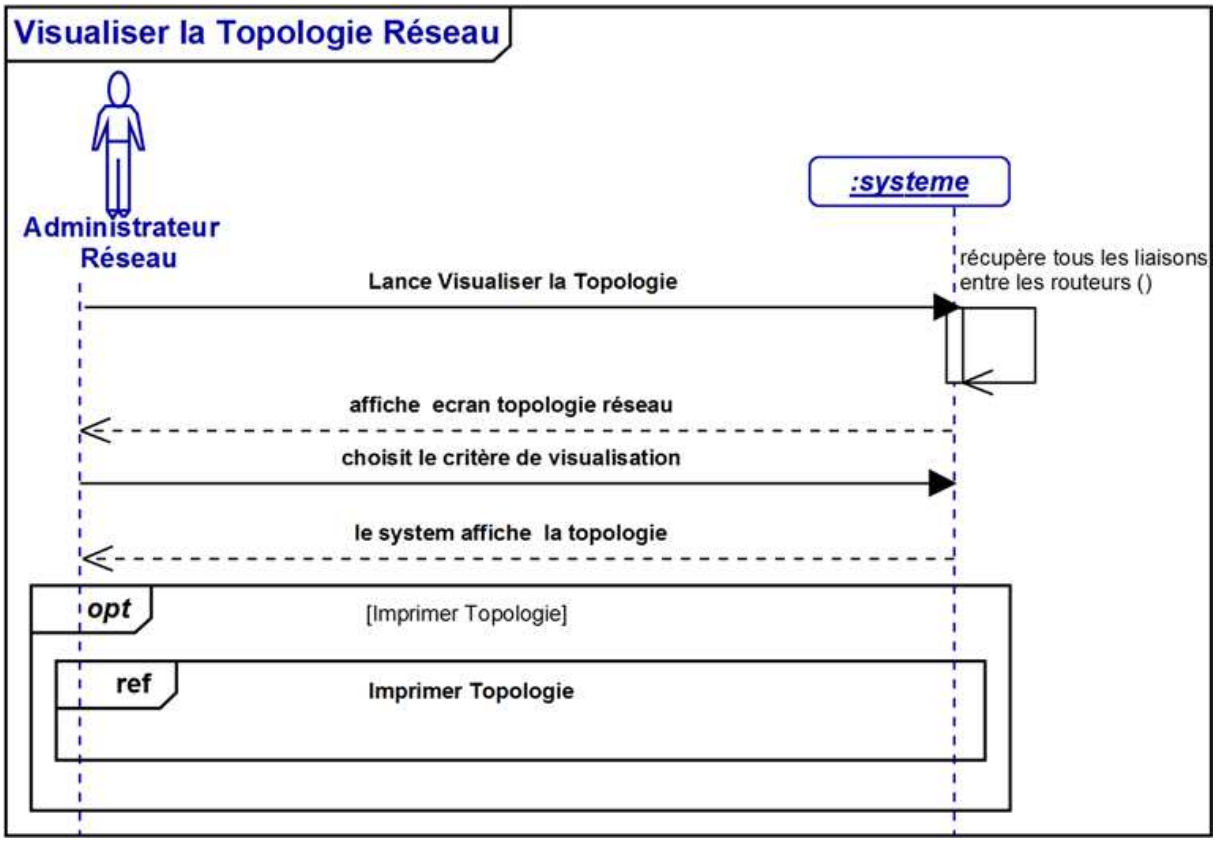

Figure 3.9 : Diagramme de séquence de Cas « Visualiser la Topologie Réseau »

❖ **Cas d'utilisation « Imprimer Topologie »**

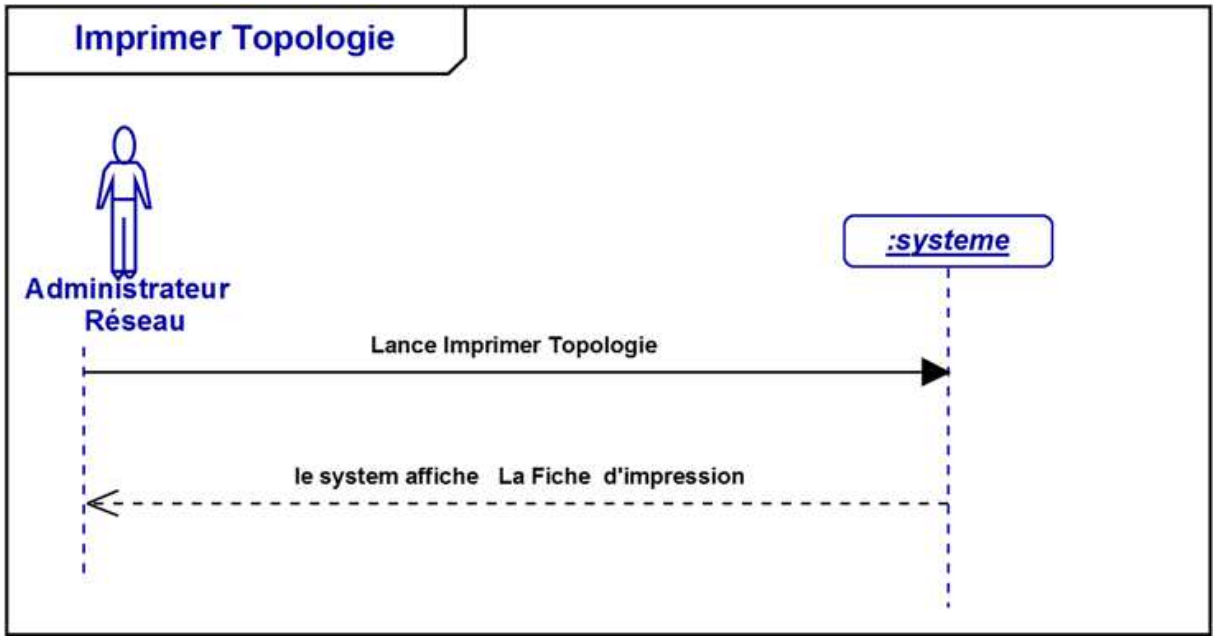

Figure 3.10 : Diagramme de séquence de « Imprimer Topologie »

III.1.3.2.Phase d'analyse

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

- **Diagramme d'activité**

A-Définition

Le diagramme d'activité représente le déroulement d'un cas d'utilisation réalisée par le système, avec tous les branchements conditionnels et toutes les boucles possibles. C'est un graphe orienté d'actions et de transitions. Les transitions sont franchies lors de la fin des actions, des étapes peuvent être réalisées en parallèle ou en séquence. Les diagrammes d'activités 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 [21].

B- Les composants de base de diagramme d'activité :

-Nœud initial : Il indique le début de déroulement de cas d'utilisation modélisée, 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 à la fin de déroulement de cas d'utilisation modélisée, 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 « Authentification »

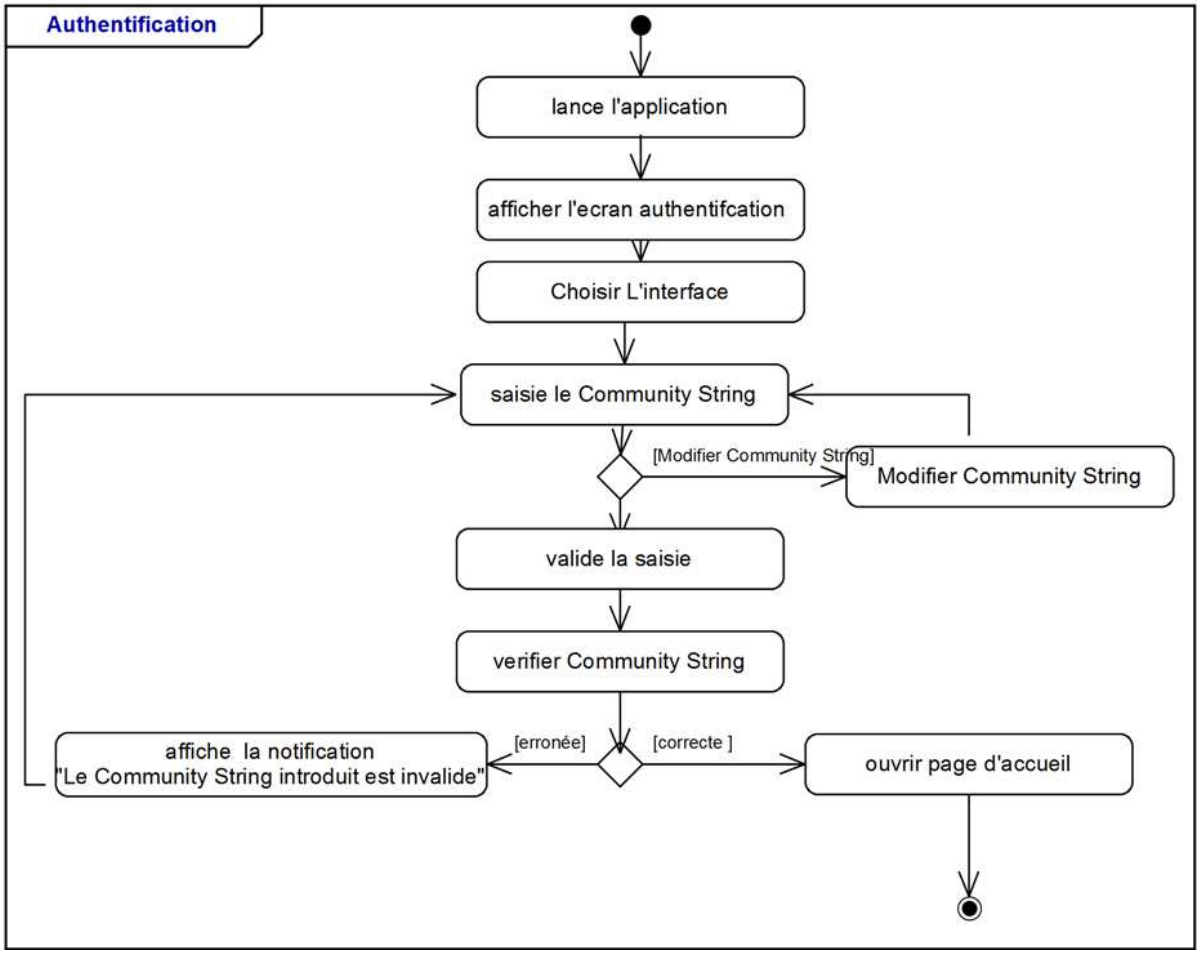

Figure 3.11 : Diagramme d'activité de Cas d'utilisation « Authentification »

❖ Cas d'utilisation « Modifier Community String »

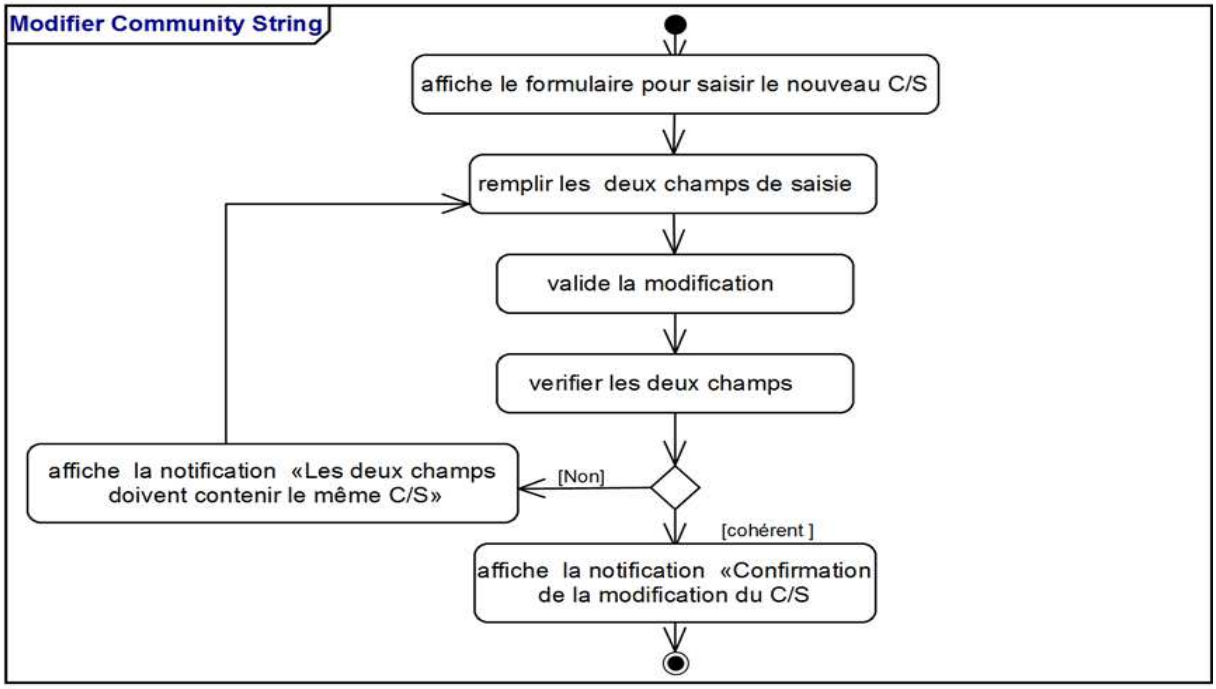

Figure 3.12 : Diagramme d'activité de Cas d'utilisation « Modifier Community String »

❖ Cas d'utilisation « Choisir L'interface »

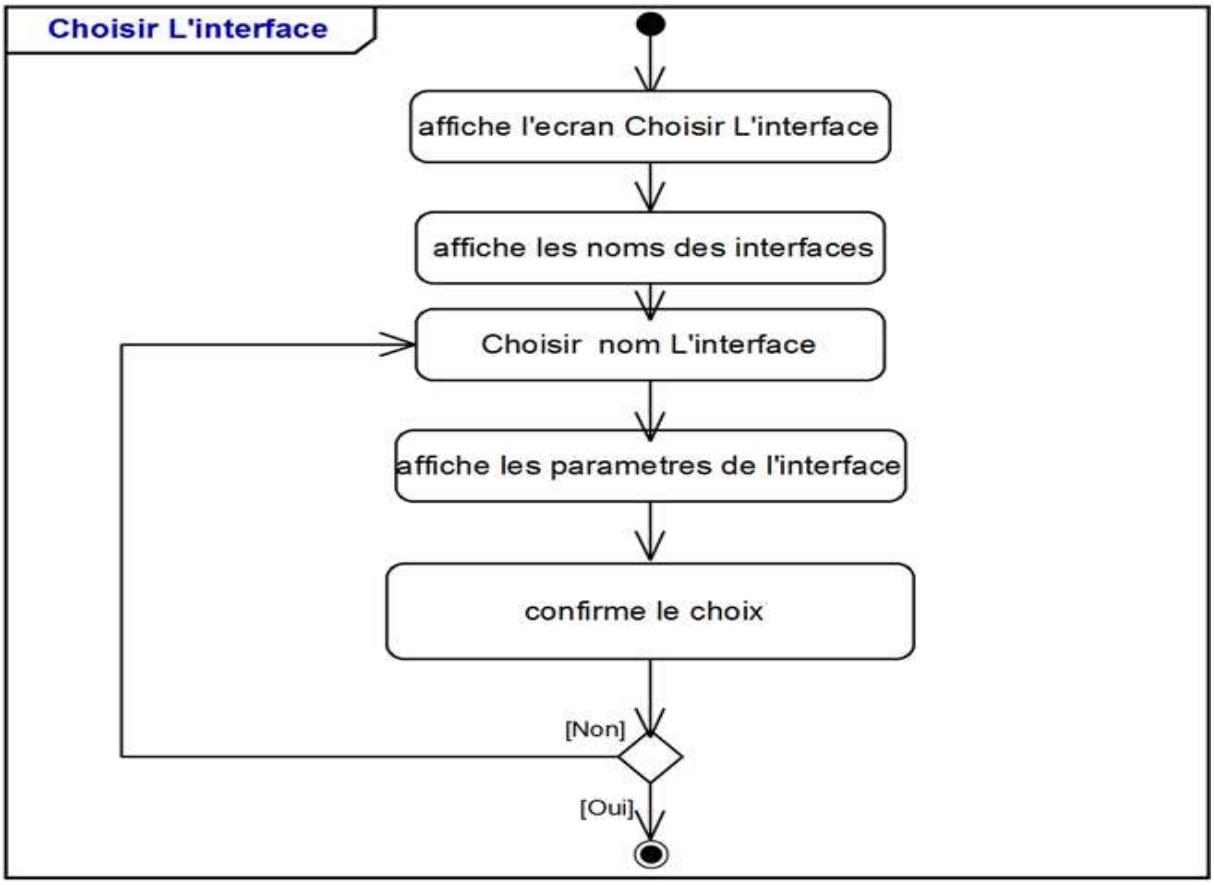

Figure 3.13 : Diagramme d'activité de Cas d'utilisation « Choisir L'interface »

- **Cas d'utilisation « Réinitialisation »**

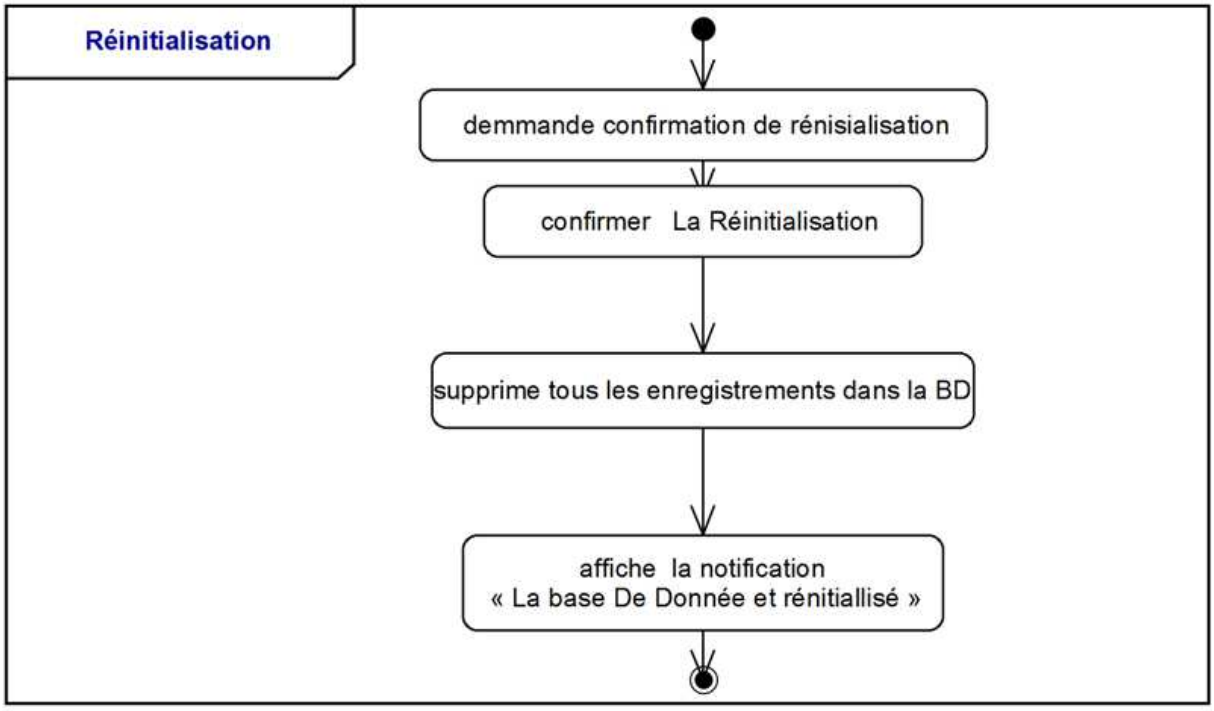

Figure 3.14 : Diagramme d'activité de Cas d'utilisation « Réinitialisation »

- **Cas d'utilisation « Auto-Découverte la Topologie Réseau »**

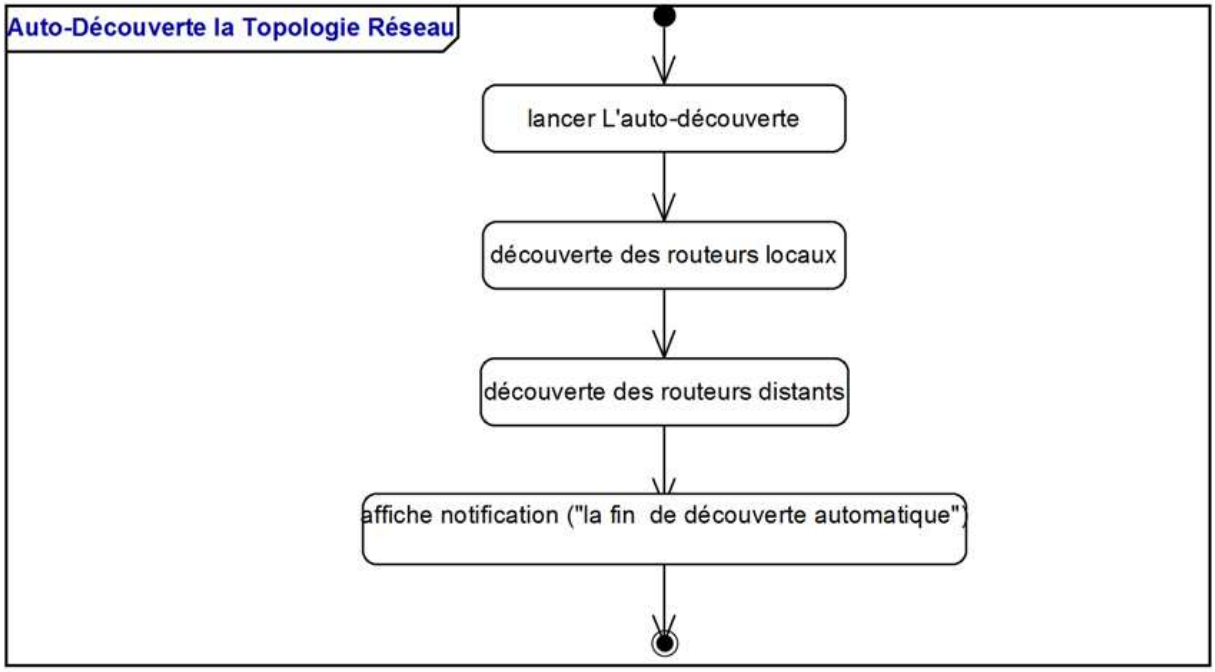

Figure 3.15 : Diagramme d'activité de Cas d'utilisation« Auto-Découverte la Topologie Réseau »

❖ Cas d'utilisation « Visualiser la Topologie Réseau »

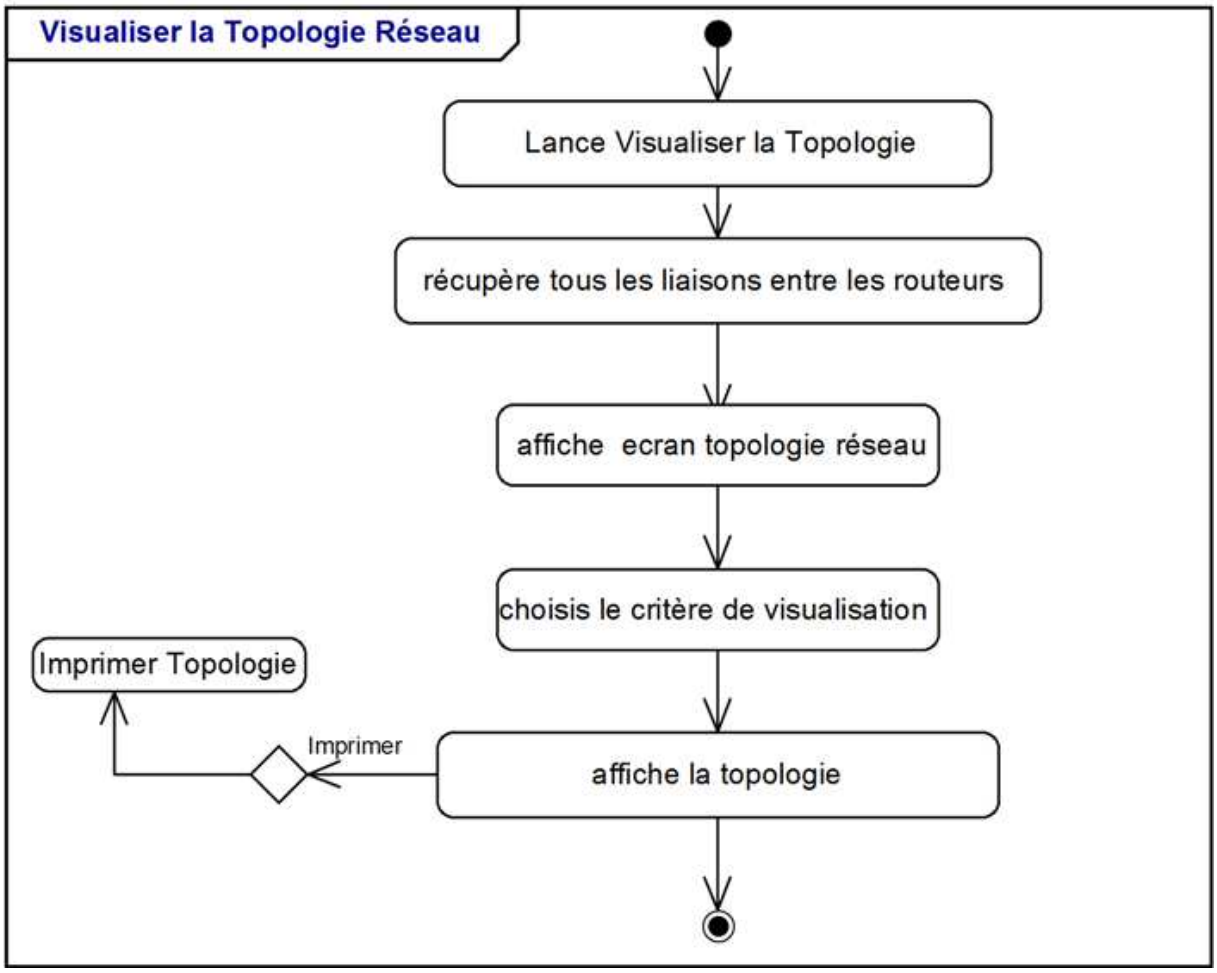

Figure 3.16 : Diagramme d'activité de Cas d'utilisation « Visualiser la topologie Réseau »

❖ Cas d'utilisation « Imprimer Topologie »

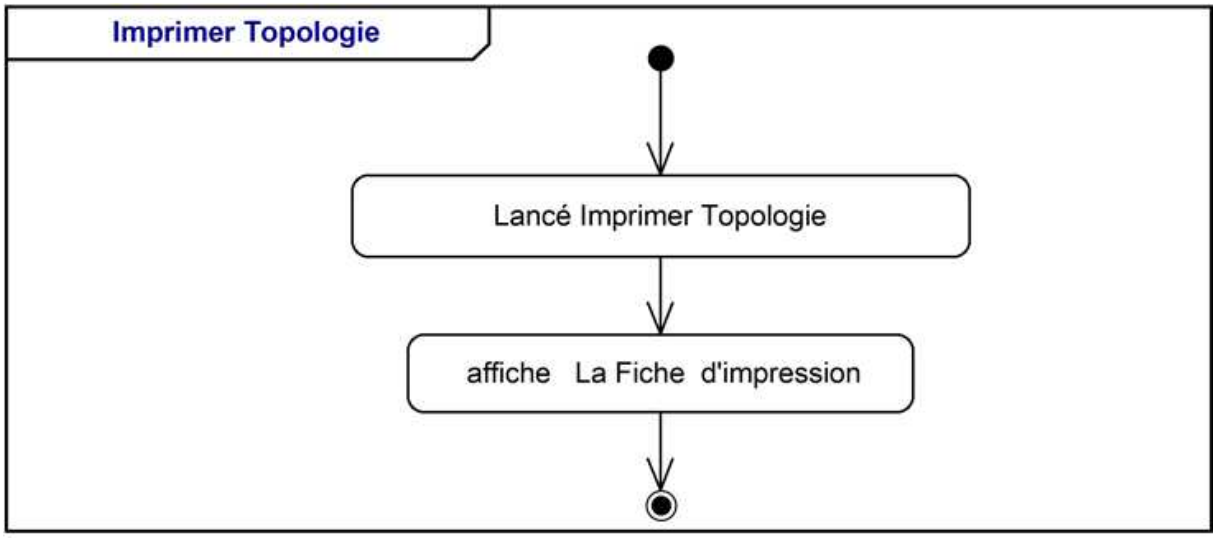

Figure 3.17 : Diagramme d'activité de Cas d'utilisation « Imprimer Topologie »

III.1.3.3.Phase de conception

La phase d'analyse est permet de décrire les objets internes du système et les interactions entre ces objets :

- **Diagrammes d'interaction**

A- Définition

Il modélise comment les objets logiciels vont interagir et communique entre eux (basé sur l'échange de messages) pour réaliser les opérations.

L'expression de diagramme d'interactions englobe principalement le diagramme de séquence et le diagramme de classe participant

Jacobson a proposé le premier des stéréotypes de classes pour décrire la réalisation d'un cas d'utilisation, pour remplacer le système vu comme une boîte noire (le diagramme de séquence) par des objets internes de logiciels [21].

B-Stéréotypes de Jacobson :

Jacobson distingue les trois stéréotypes suivants :

- « boundary » : classes qui servent à modéliser les interactions entre le système et ses acteurs.
- « control » : classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.
- « entity » : classes qui servent à modéliser des informations durables et souvent persistantes.

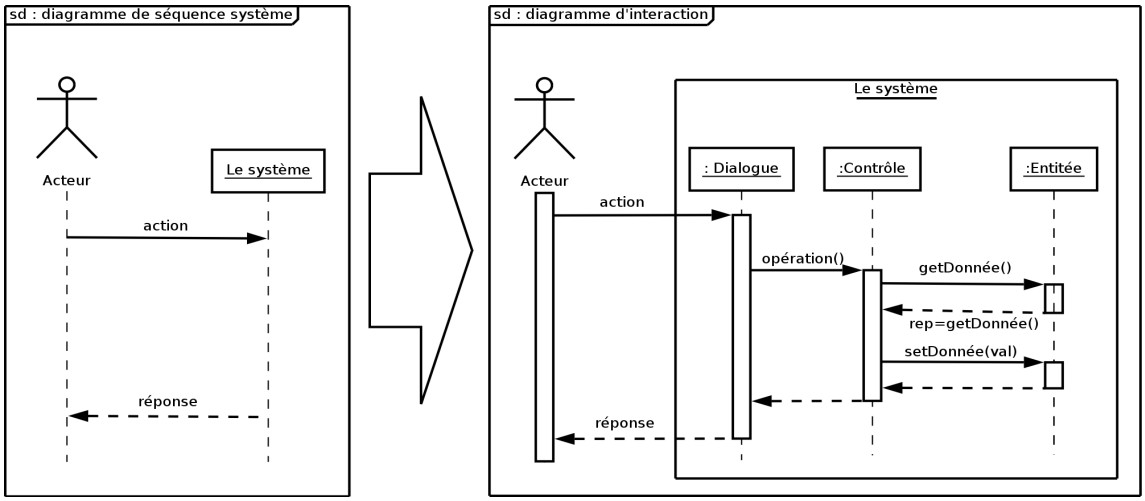

Figure 3.18 : passage de diagramme de séquence vers le diagramme global d'interaction

❖ Cas d'utilisation «Authentification»

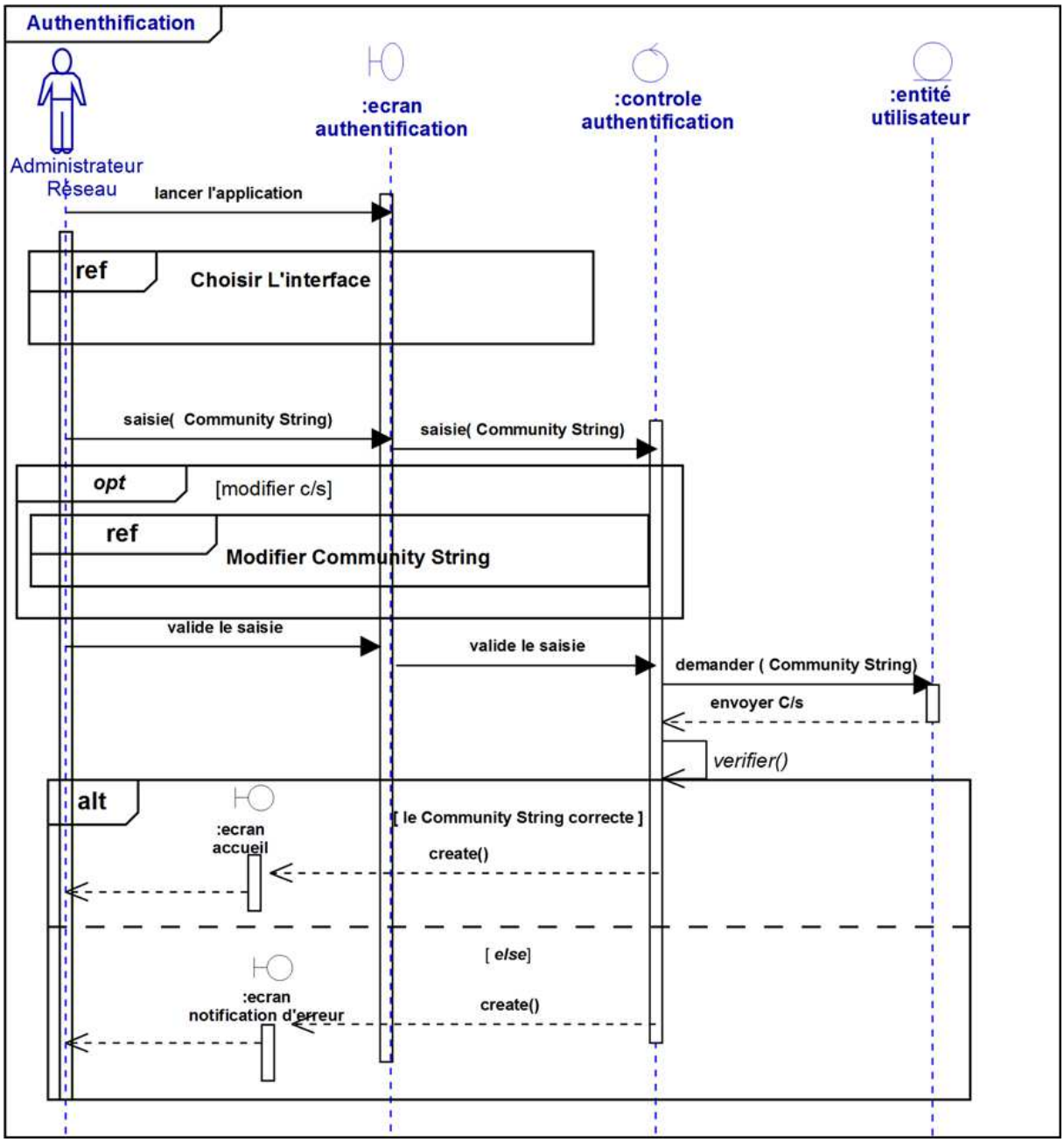

Figure 3.19 : Diagramme d'interaction de Cas d'utilisation «Authentification»

❖ Cas d'utilisation «Modifier Community String»

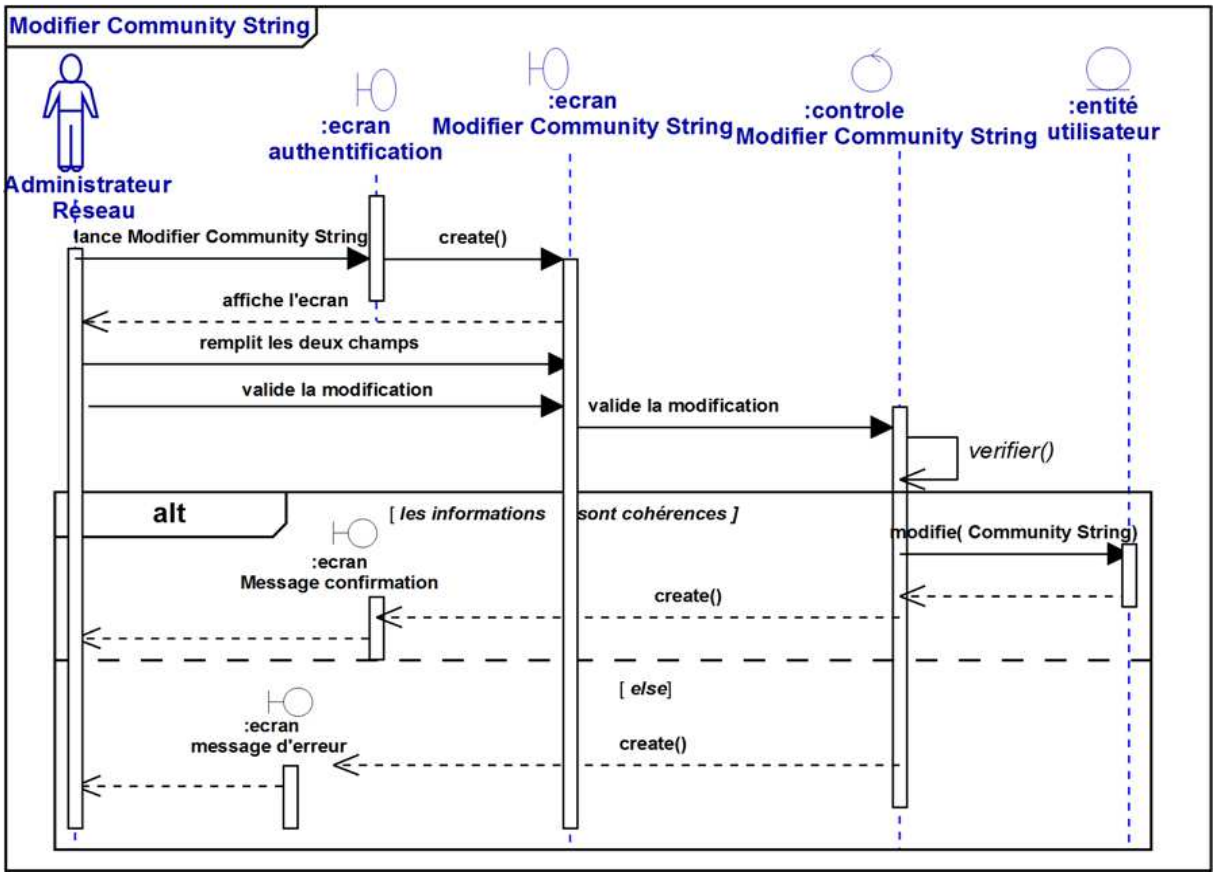

Figure 3.20 : Diagramme d'interaction de Cas «Modifier Community String»

❖ Cas d'utilisation «Choisir L'interface»

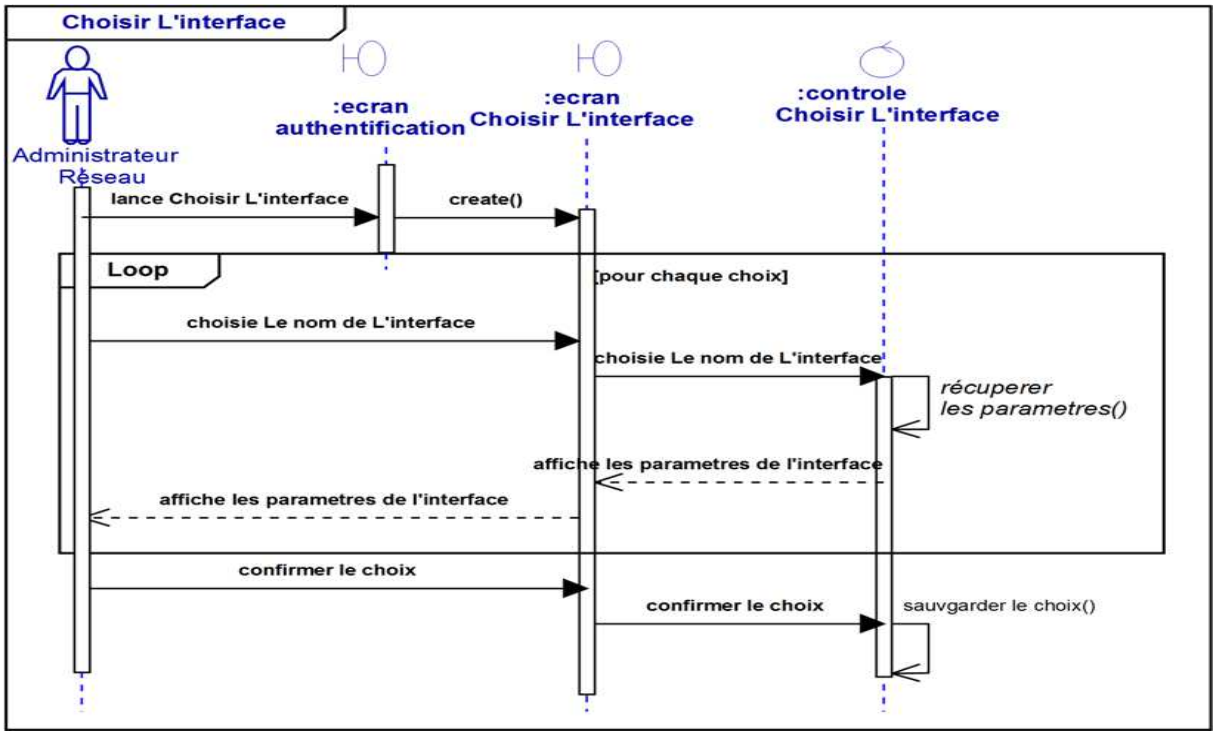

Figure 3.21 : Diagramme d'interaction de Cas «Choisir L'interface»

❖ Cas d'utilisation «Réinitialisation»

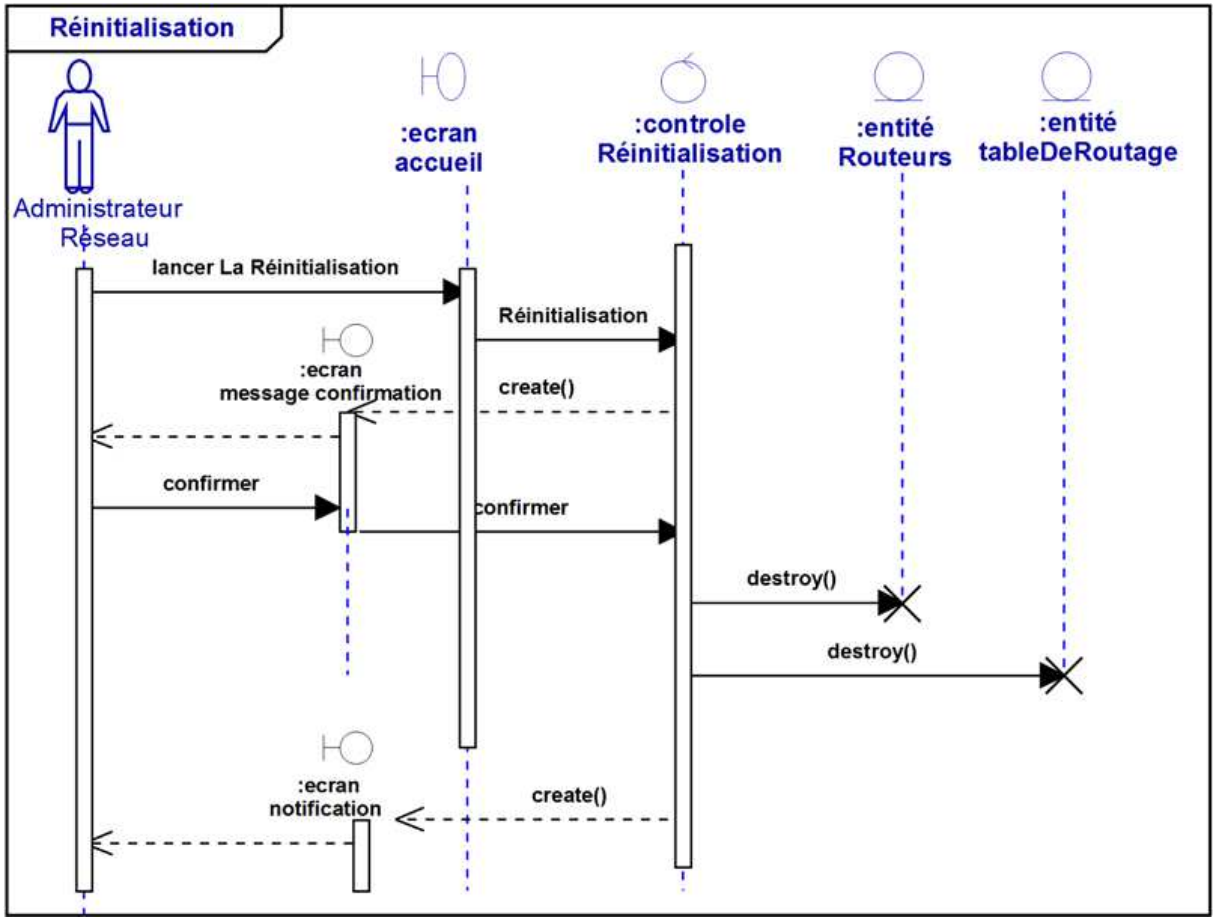

Figure 3.22 : Diagramme d'interaction de Cas «Réinitialisation»

❖ Cas d'utilisation «Auto-Découverte la Topologie Réseau»

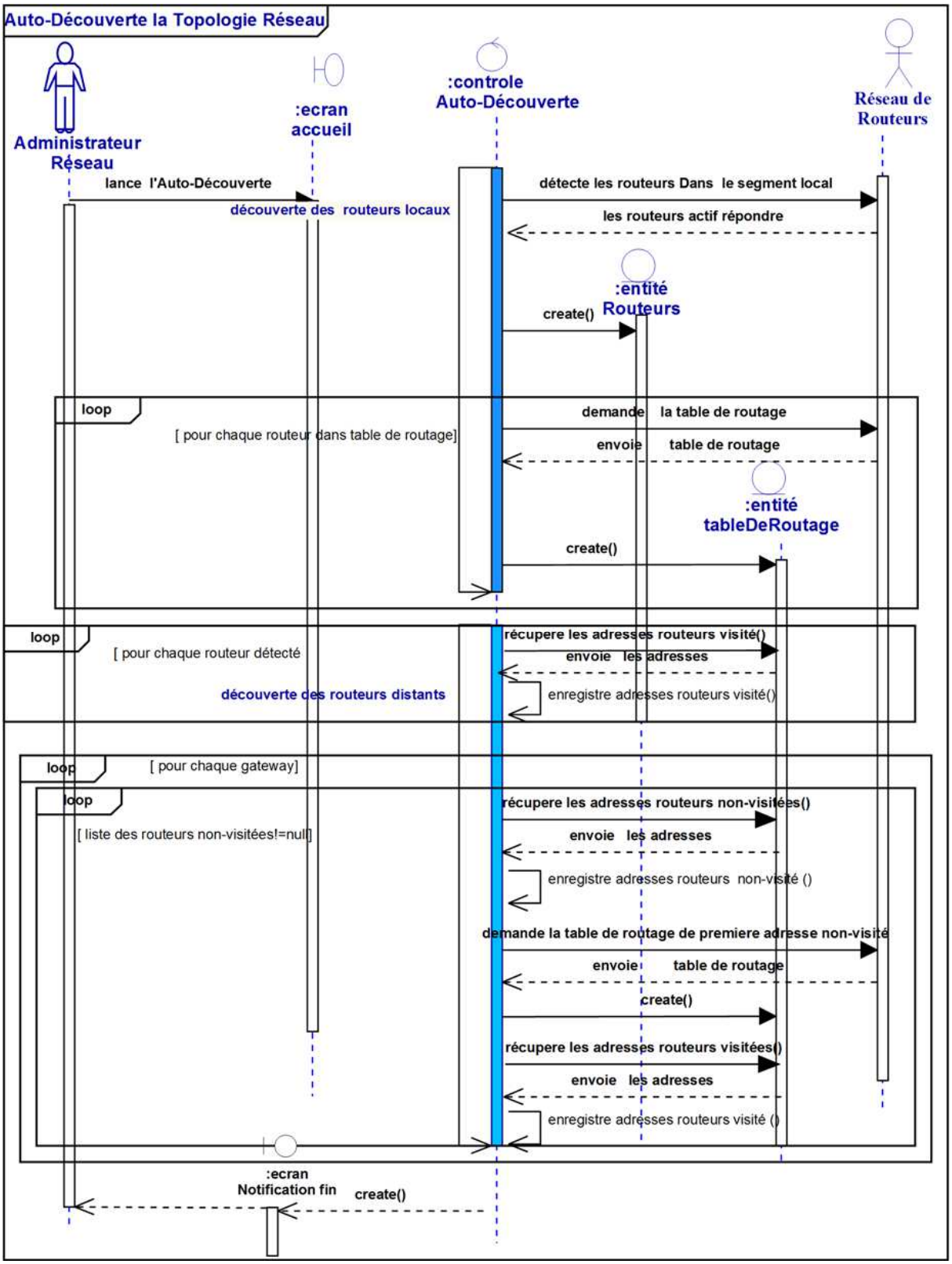

Figure 3.23 : Diagramme d'interaction de Cas «Auto-Découverte la Topologie Réseau»

❖ Cas d'utilisation «Visualiser la Topologie Réseau»

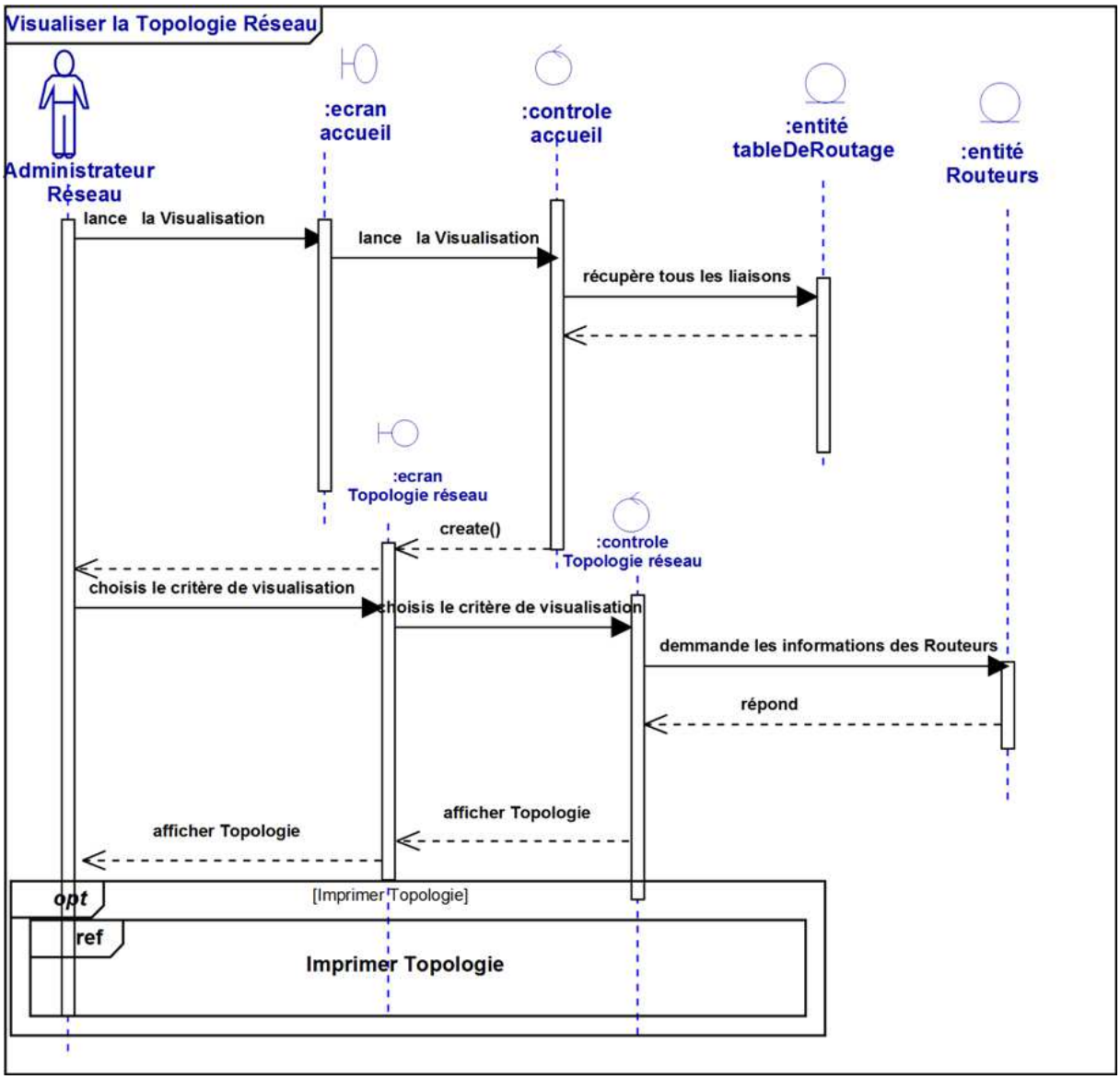

Figure 3.24 : Diagramme d'interaction de Cas «Visualiser la Topologie Réseau»

❖ **Cas d'utilisation «Imprimer Topologie»**

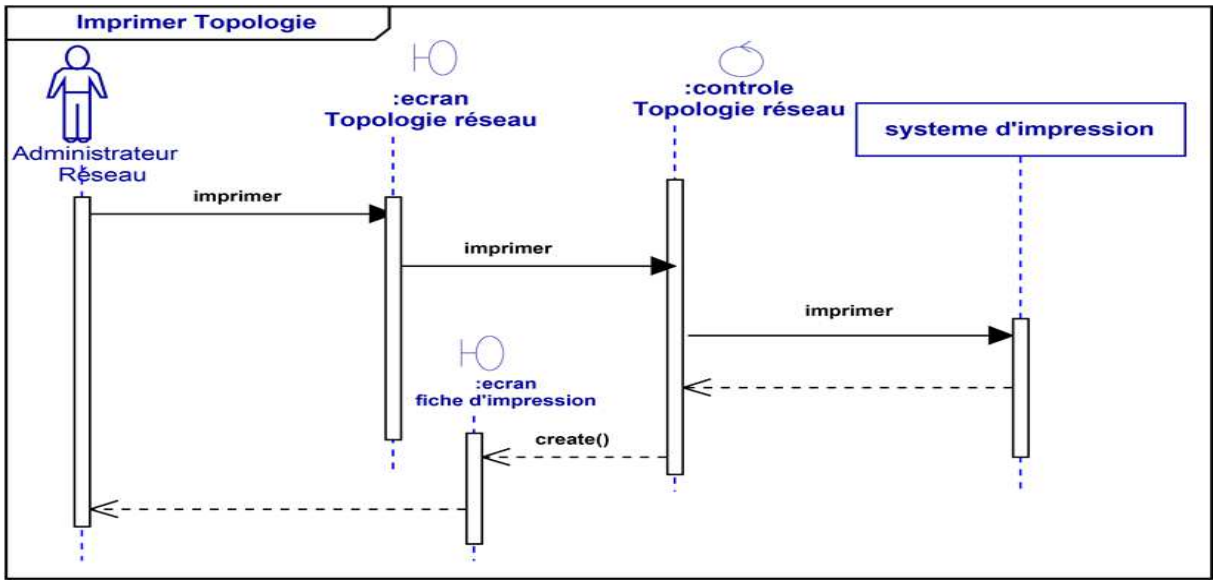

Figure 3.25 : Diagramme d'interaction de Cas «Imprimer Topologie »

➢ Diagramme de classe

A-Définition

Il est considéré comme le plus important de la modélisation orientée objet, ce diagramme représente la vue statique du système, le diagramme de classe est un diagramme entité_association identifie la structure des classes d'un système, y compris les propriétés et les méthodes de chaque classe ainsi que les différentes relations entre celles-ci.

B- Les composants de base de diagramme de classe :

- Les classes : Sont les modules de base de la programmation orientée objet. Une classe est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe. La section centrale définit les propriétés de la classe. La section du bas énumère les méthodes de la classe.

- L'association : représente une relation sémantique durable entre deux classes. Elle est modélisée par une ligne reliant les deux classes. Cette ligne peut être qualifiée avec le type de relation, et peut également comporter des règles de multiplicité (par exemple un à un, un à plusieurs, plusieurs à plusieurs) pour la relation.

-La composition : Si une classe ne peut pas exister par elle-même, mais doit être un membre d'une autre classe, alors elle possède une relation de composition avec la classe contenant. Une relation de composition est indiquée par une ligne avec un "diamant" rempli. Une composition est une agrégation plus forte impliquant que :

- un élément ne peut appartenir qu'à un seul agrégat composite (agrégation non partagée) ;
- la destruction de l'agrégat composite entraîne la destruction de tous ses éléments(le composite est responsable du cycle de vie des parties).

-L'agrégation : indique une relation de contenance, Elle décrite par une relation "possède". Une relation d'agrégation est représentée par une ligne avec un "diamant" creux.

-La généralisation : est l'équivalent d'une relation d'héritage en terme orienté objet. Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe "parent".

-La dépendance : La dépendance entre deux classes permet de représenter l'existence d'un lien sémantique. 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 en pointillé

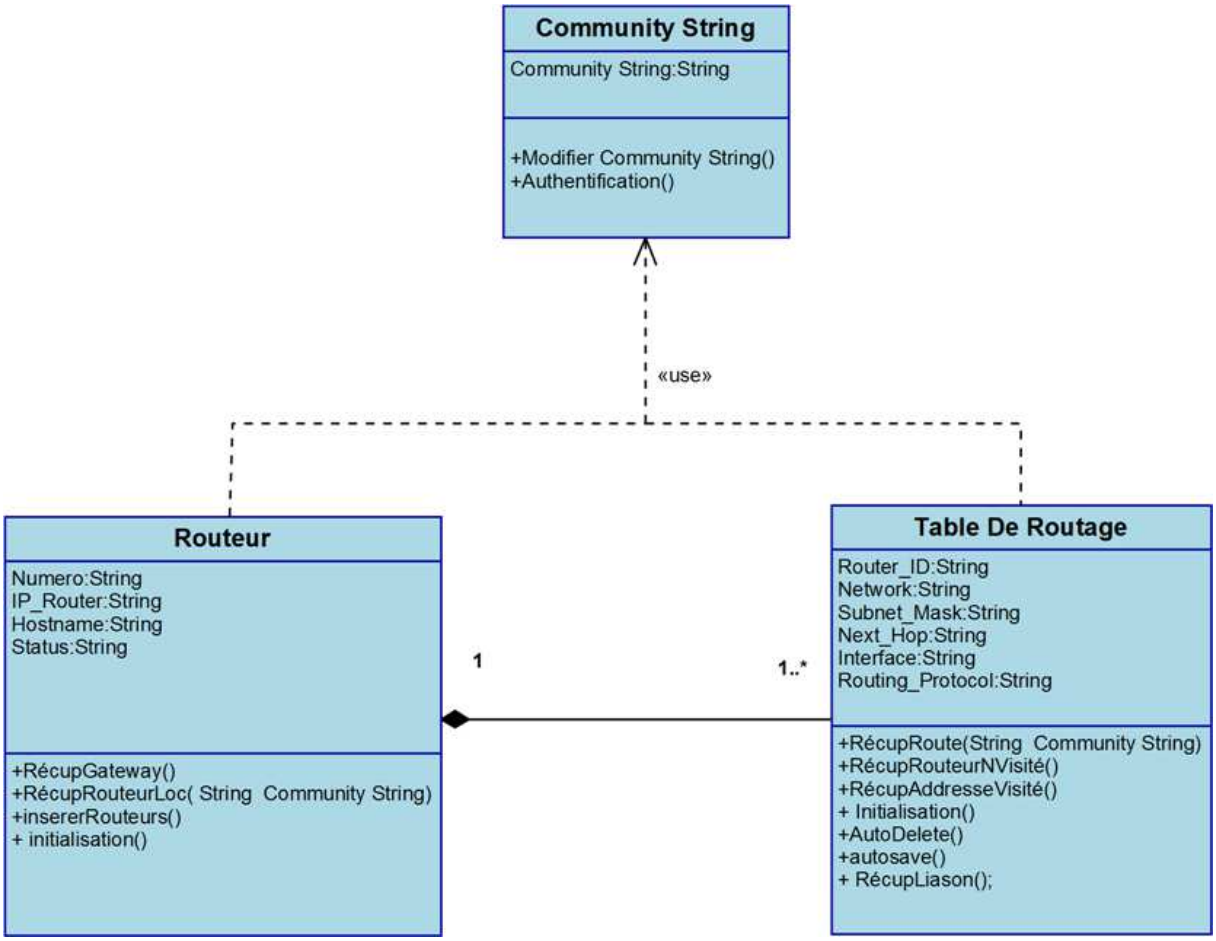

Figure 3.26 : diagramme de classe

- **les règles de gestion:**
  - Chaque routeur est composé d'une ou plusieurs routes dans la table de routage.
  - Chaque route dans la table de routage n'appartient qu'à un seul routeur.
  - Relation de dépendance :
    - classe Table de routage utilisé Community String comme argument dans la signature de L'opération RécupRoute (String Community String).
    - classe Routeurs utilisé Community String comme argument dans la signature de L'opération RécupRouteurLoc (String Community String)

III.2. Réalisation

Dans cette partie, nous présentons l'environnement de développement de notre système.

III.2.1. Netbeans :

NetBeans est un environnement de développement intégré (EDI), open source, très utile qui permet le développement en java. NetBeans permet également de supporter une large variété de langages de programmation telle que C, C++, JavaScript, XML, PHP et HTML.

Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, éditeur graphique d'interfaces et de pages Web). NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing [Java]).

III.2.2.Le langage de programmation JAVA

Java est un langage de programmation orienté objet. Il est très utilisé dans le domaine de développement créé par James Gosling et Patrick Naughton, employés de Sun Microsystems.Il est fourni avec un ensemble d'outils (le JDK Java Development Kit) et un ensemble de packages : ensemble de classes. Ces différentes classes de base couvrent beaucoup de domaines (entrées/sorties, interface graphique, réseau, etc.) Cette richesse en "librairies standards" explique sûrement en partie le succès de Java.

La particularité principale de java est que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que : Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java [22].

III.2.3.Présentation de JDK :

JDK Java Développent Kit est nécessaire pour développer des applications Java. Le JDK contient le compilateur Java et les bibliothèques de programmation nécessaires à l'élaboration de programmes Java.

III.2. 4.Présentation de JDMK

JDMK Java Dynamic Management Kit : est une technologie Java qui fournit une API Java et une collection d'outils logiciels pour le développement et la conception d'applications de base JMX (Java Management Extensions). Il fournit un ensemble de classes Java et des outils qui permettent de développer facilement des solutions de surveillance et de gestion sécurisées basées sur la JMX spécifications et les sur les normes SNMP.

III.2. 5.JUNG:

(Java Universel Network/Graph) est une bibliothèque graphique gratuite qui fournit un langage commun et extensible pour la modélisation des objets, des algorithmes de la théorie des graphes, et l'analyse et la visualisation de données qui peut être représentées sous forme de graphique ou d'un réseau. Il est écrit en Java, qui permet aux applications basées sur JUNG à faire usage des vastes capacités intégrées de l'API Java. L'architecture JUNG est conçue pour soutenir une variété de représentations d'entités et leurs relations, comme graph dirigé et non dirigé et d'autres représentations [23].

III.2.6.Access

Access (officiellement Microsoft Office Access) est un SGBD relationnel édité par Microsoft. Il fait partie de la suite bureautique MS Office, fonctionnant pour le système d'exploitation Microsoft Windows.

MS Access est un logiciel utilisant des fichiers au format Access (extension de fichier mdb pour Microsoft DataBase (extension *.accdb depuis la version 2007). Il est compatible avec les requêtes SQL (sous certaines restrictions) et dispose d'une interface graphique pour saisir les requêtes. Il permet aussi de configurer, avec des assistants ou librement.

Comme beaucoup de systèmes de gestion de bases de données relationnelles, ses données peuvent être utilisées dans des programmes écrits dans divers langages.

III.2. 7.ODBC

Open Database Connectivity (ODBC) est une interface stratégique conçue par Microsoft pour accéder et manipuler plusieurs des bases de données dans un environnement hétérogène de Systèmes de Gestion de Bases de Données (SGBD) relationnel set non relationnels.

ODBC comporte un registre des bases de données (source de données) disponibles depuis l'ordinateur de l'usager, une interface graphique permet à l'usager d'ajouter des bases de données au registre. ODBC affranchit les éditeurs de logiciels et les développeurs de l'apprentissage de multiples interfaces de programmation d'applications(API). ODBC fournit maintenant une interface universelle d'accès aux données. Avec ODBC, les applications peuvent voir, éditer et modifier des données de nombreuses et multiples bases de données.

Le logiciel ODBC de Microsoft est fourni avec les pilotes pour les SGBD Access, FoxPro et SQL Server, du même auteur. De nombreux éditeurs de SGBD (Oracle, MySQL...) fournissent les pilotes pour leurs produits.

III.2.8.Le Simulateur GNS3

GNS3 (Graphical Network Simulator) est un simulateur de réseau graphique qui permet l'émulation des réseaux complexes. GNS3 permet d'émulation à l'aide de Cisco Internetwork Operating Systems. Il vous permet d'exécuter un IOS Cisco dans un environnement virtuel sur votre ordinateur. GNS3 est une interface graphique pour un produit appelé Dynagen. Dynamips est le programme de base qui permet l'émulation d'IOS. Dynagen s'exécute au-dessus de Dynamips pour créer un environnement plus convivial, basé sur le texte environnement. Un utilisateur peut créer des topologies de réseaux complexes [24].

III.3.Les interfaces de système « SpiderNet » :

III.3.1. Interface Authentification

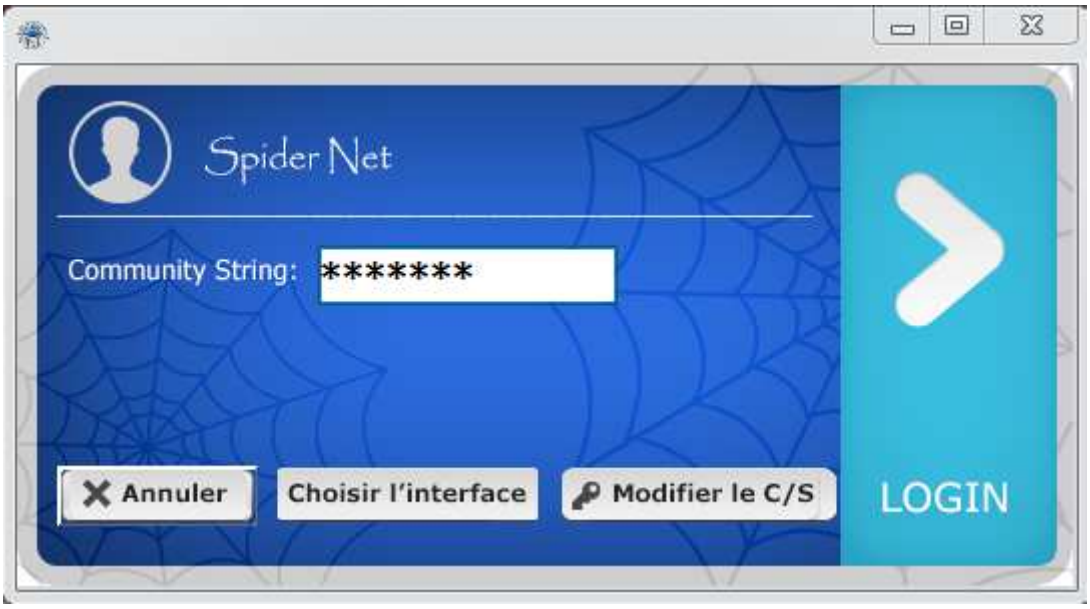

Figure 3.27 : Interface Authentification

III.3.2. Interface Modifier Community String

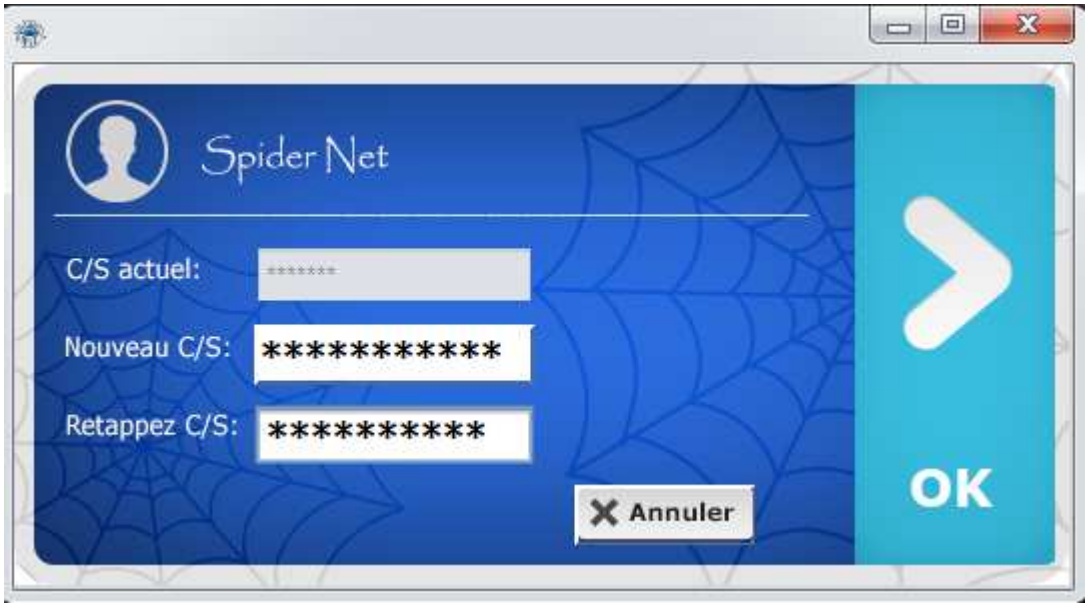

Figure 3.28 : Interface Modifier Community String

III.3.3. Interface Choisir L'interface

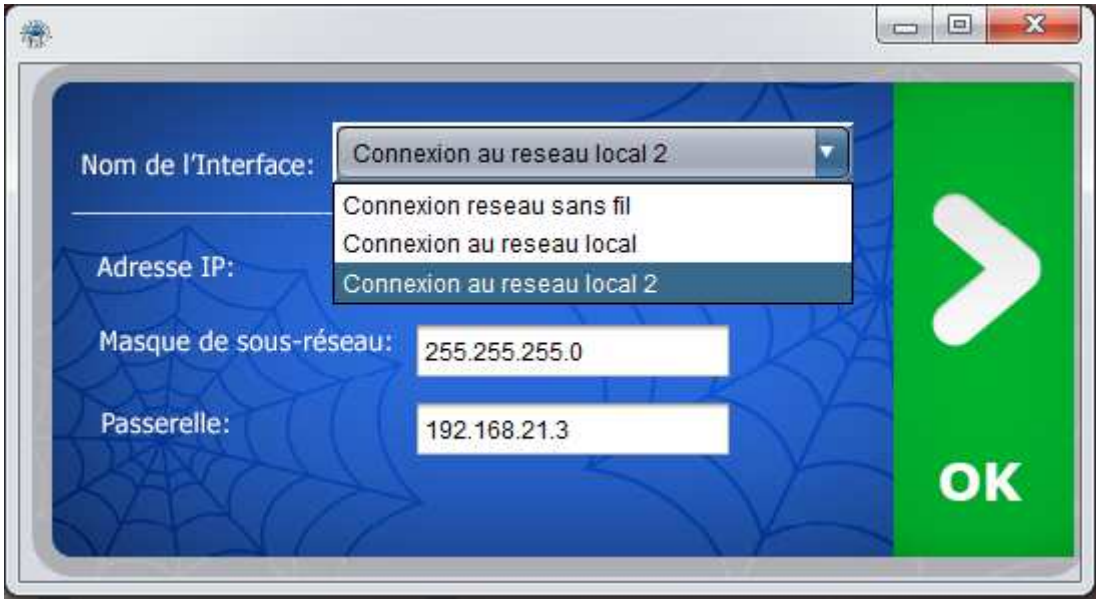

Figure 3.29 : Interface Choisir L'interface

III.3.4. Interface page d'accueil

Figure 3.30 : Interface page d'accueil

III.3.5. Interface Visualiser la Topologie Réseau

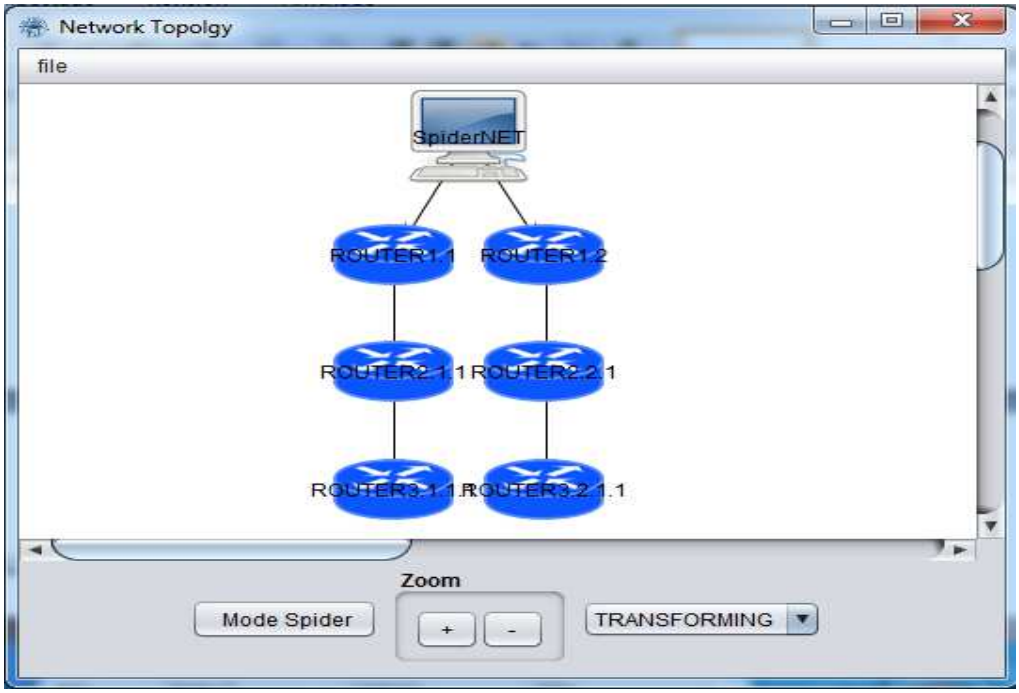

Figure 3.31 : Interface Visualiser la Topologie Réseau

Conclusion

Dans ce chapitre nous avons présenté une conception détaillée de notre projet à travers un ensemble de diagramme UML, modélisant le comportement de **«SpiderNet »**. Cette phase nous a permis de concrétiser l'étude théorique par un logiciel pouvant être pratiquement exploité au niveau des entreprises, répondant aux besoins des administrateurs et optimisant les fonctionnalités des systèmes de gestion réseau.

Conclusion générale

Conclusion générale

L'objectif de notre projet était la réalisation d'un système permettant la découverte automatique des topologies réseau, offrant à tout administrateur réseau au niveau d'un organisme ou bien entreprise étendue un outil indispensable dans la prise des décisions, assurant la fiabilité des systèmes de gestion et permettant un meilleur contrôle sur la convergence ainsi que les futures extensions des réseaux.

Notre solution s'intéresse plus à la perception humaine, en représentant une masse importante d'informations sous forme graphique multicritères et interactive, bien adaptée aux besoins des administrateurs garantissant une meilleure compréhensibilité, une simplicité d'accès aux informations et une intégration rapide pour les débutant dans le domaine d'administration des réseaux.

Notre application nommée **« SpiderNet »** a été testée sur différents scénarios simulés utilisant GNS3, couvrant la majorité des plans réseaux les plus utilisés de nos jours dans les entreprises à large échelle. Pour des résultats optimaux **« SpiderNet »** doit être exploité dans le HQ (quartier général) des entreprises, là ou le routage vers tous les réseaux des branches est centralisé.

Durant l'étude du système **« SpiderNet »** nous étions concentrés sur la garantie de couverture de tout le réseau et la meilleure façon de visualiser la topologie**.** Nous projetons dans des futurs travaux l'optimisation de l'algorithme de découverte afin d'obtenir des résultats dans les plus brefs délais tout en garantissant la sécurité des informations de gestion parcourant le réseau.

Liste des abréviations

ARP : Address Resolution Protocol

ASN.1 : Abstract Syntax Notation One

CDP : Cisco Discovery Protocol

CMIP : Common Management Information Protocol

GNS 3 : Graphical Network Simulator

ICMP : Internet Control Message Protocol

IETF : Internet Engineering Task Force

IP : Internet Protocol

ISO : International Organisation for Standardisation

MIB : Management Information Base

NMS : Network Management Station

OID : Object Identifier

OSI : Open System Interconnection

SNMP : Simple Network Management Protocol

TCP : Transmission Control Protocol

RFC : Request for comments

UDP : User Data Protocol

JUNG : Java Universel Network/Graph

JDMK : Java Dynamic Management Kit

JDK : Java Développent Kit

JDBC : Java DataBase Connectivity

LLDP: Link Layer Discovery Protocol

UML : Unified Modeling Language

ODBC : Open DataBase Connectivit

Références bibliographiques

[1] Advances in Network Management, Jianguo Ding, 2010

[2] Diagnostic décentralisé de systèmes a événements discrets : application aux réseaux de télécommunications, Yannick PENCOLE.

[3] Pratique de la gestion de réseau, Nazim Agoulmine Omar Cherkaoui, Groupe Eyrolles, 2003.

[4] Managing Internetworks with SNMP, Mark A, Miller, 1997.

[5] Opération SNMP gestion de réseaux et services une introduction, Daniel Ranc.

[6] Gestion de réseaux IP, Stefano Ventura Markus Jaton.

[7] Computer Networks: An Open Source Approach, Ying-Dar Lin, Ren-Hung Hwang, 2010.

[8] SNMP architecture : les protocoles pour la gestion des réseaux informatiques, Victor MORARU, Hanoi, juillet 2005.

[9] Gestion de réseaux et services, Daniel Ranc, 2004.

[10] La gestion réseau et le protocole SNMP, Aurélien Méré.

[11] Essential SNMP Douglas R. Mauro and Kevin J. Schmidt Beijing, 2005.

[12] Network Topology Discovery Algorithm based on SNMP and ICMP ,Zhiming Li, 2013.

[13] Découverte et agrégation de topologies de réseaux, Walid Htira, 2008.

[14] Study on network topology discovery in ip networks, Yao Zhao, Jianliang Yan, 2010.

[15] Analysis and Research of Network Topology Discovery Method, He Peng, Gu Xiang, 2010.

[16] Topology Discovery Using Cisco Discovery Protocol, Sergio R. Rodriguez, 2009

[17] Network Discovery Protocol LLDP and LLDP-MED, Prof. Vahida Z. Attar, Piyush

[18] Le langage de modélisation objet UML, Olivier Guibert, 2004.

[19] UML 2 par la pratique, Pascal Roques, 2006.

[20] UML 2 Modéliser une application web, Pascal Roques, 2007.

[21] UML 2 Analyse et conception, Joseph Gabay, David Gabay, 2008.

[22] Développons en Java Par: Jean Michel DOUDOUX.

[23] http://jung.sourceforge.net

[24] http://www.gns3.net