République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abdrahmane Mira de Bejaia

Faculté des Sciences exacte

Département d'informatique

En vue de l'obtention d'un diplome de Master Profitionnel en Informatique

Implémentation d'une application de gestion réseau basée sur le protocole SNMP

Réalisé par :

M^elle *ASLI Fatma*

M^r *GHILAS Massinissa*

Présidente	M^elle	K.GHIDOUCHE	M.A.B	U. A/Mira Béjaia.
Promotrice	M^elle	M.GAGAOUA	M.A.B	U.A/Mira Béjaia.
Co-promoteur	M^r	H.GHILAS	M.A.B	U.A/Mira Béjaia.
Examinatrices	M^elle	S.TOULOUM	M.A.B	U. A/Mira Béjaia.
	M^elle	S.GHANEM	M.A.B	U. A/Mira Béjaia.

Année Universitaire 2012 – 2013

Louange A Dieu, le miséricordieux, sans Lui rien de tout cela n'aurait pu être.

Nous tenons tout d'abord à remercier M^{elle} M.GAGAOUA ,pour l'honneur qu'elle nous a fait en acceptant de nous encadrer. ses conseils précieux ont permis une bonne orientation dans la réalisation de ce modeste travail.

Nous tenons également à remercier M^{r} H.GHILAS pour son aide et ses précieux conseils.

Nous remercions M^{elle} K.GHIDOUCHE, M^{elle} S.TOULOUM et M^{elle} S.GHANEM d'avoir accepté de faire partie du jury et d'avoir consacré leurs temps à la lecture et à la correction de ce mémoire.

Sans oublier M^{r} A.AKILAL, qui nous a soutenu tout au long de la réalisation de ce projet .

Nos remerciements particuliers à M^{r} S.FAOUDI et M^{r} A.OUATAH qui nous a proposé le sujet et M^{r} A.HEMITERI qui nous a aorienté vers la SONATRACH.

Nos remerciements les plus vifs vont tout particulièrement à nos parents.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Je dédie ce modeste travail :

A mes chères parents.

A mes sœurs : Fazia et Fatima, à mon cher frère Mazigh .

A mes amis et mes collègues.

Mssinissa

Je dédie ce modeste travail :

Aux personnes qui sont chères à mon cœur.

Fatma

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

* **ICMP :** Internet Contrôle Message Protocol.
* **IP** : Internet Protocol.
* **IAB** : Internet Activities Board.
* **TCP** : Transmission Contrôle Protocol.
* **HEMS** : High-Level Entity Management System.
* **HMP** : Host Monotoring Protocol.
* **CMIS** : Commun Management Information Semice.
* **CMIP** : Commun Management Information Protocol.
* **PDU** : Protocol Data Unit.
* **SNMP** : Simple Network Management Protocol.
* **SGMP** : Simple Gateway Management Protocol.
* **ISO** : International Organization for Standardization.
* **MIB** : Management Information Base.
* **ITEF** : Internet Engineering Task Force.
* **RFC** : Request For Comments.
* **SMI** : Structure of Management Information.
* **NMS** : Network Management Station.
* **MIT** : Management Information Tree.
* **DOD** : Departement Of Defense.
* **ASN** : Abstract Syntax Notation.
* **OID** : object Identified.
* **UDP** : User Datagram Protocol.
* **USM** : User-based Security Model.
* **VACM** : View-based Access Control Model.
* **SNMPv1** : Simple Network Management Protocol version 1.

* **SNMPv2** : Simple Network Management Protocol version 2.
* **SNMPv3** : Simple Network Management Protocol version 3.
* **UML** : Unified Modeling Language.
* **UP** : Unified Processus.
* **DRGB** : Direction Régionale de Bejaia.

Introduction générale

La gestion d'un réseau est définie comme étant le processus de contrôle d'un réseau de données de façon à maximiser son efficacité et améliorer ses performances. Le processus de gestion des réseaux inclut généralement la collecte des données, faite automatiquement ou via l'analyste responsable du réseau, le traitement de ces données et, ensuite, leur présentation.

La gestion de réseau recouvre de nombreuses opérations, telles que l'initialisation des paramètres de configuration du système, la gestion des erreurs, les statistiques, les diagnostics, la gestion des alarmes et leur rapport, la reconfiguration, la gestion des ressources, la sécurité, ...etc.

Pour interconnecter deux systèmes de gestion, une norme doit être respectée, et la norme la plus utilisée est SNMP (Simple Network Management Protocol).

Comme son nom l'indique le protocole **SNMP**, simple network management protocole (*protocole de gestion de réseau simplifié*) que nous allons étudier plus en détails au cours de ce travail, a pour rôle exclusif la gestion de réseau, il a été développé pour apporter des moyens simples d'administration à distance aux administrateurs.

Pour la réalisation de cette tâche, notre choix s'est porté sur le Processus **UP** ; en effet, **UP** est une solution de développement logiciel adaptée à tout type de projet.

Le langage de modélisation que nous allons utiliser est **UML** (***U****nified* ***M****odeling* ***L****anguage*), qui est une partie intégrante de la démarche **UP**, ses diagrammes sont largement utilisés dans chaque étape.

Pour l'implémentation, le choix du langage de programmation a été dicté par le type de l'application qui devrait être réalisée, ainsi, le choix s'est porté sur le langage de programmation JAVA pour les avantages qu'il offre.

La suite de ce memoire est organisé comme suit :

– **Chapitre 1** : Présente une introduction aux concepts de base de la gestion de réseau ;

il permet de comprendre les enjeux stratégiques de la gestion et de se familiariser avec ses activités.

- **Chapitre 2** : Intitulé « étude du protocole SNMP » il décrit le protocole SNMP, et explique son fonctionnement.
- **Chapitre 3** : Présente l'analyse des besoins et conception, dont nous allons décrire les différents cas d'utilisations qui seront accompagnés par des diagrammes de séquence d'une part, d'autre part nous allons établire le diagramme de classe.
- **Chapitre 4** : Présente l'imlémentation et testes qui sera réserver pour la présentation des outils de développement, ainsi que la présentation de l'application et les testes.

1

Concepts de base de gestion des réseaux Informatiques

Introduction

Nous débuterons notre étude par des généralités sur la gestion des réseaux Informatique et ses concepts de base, dont nous diviserons ce chapitre en deux parties :

La première partie sera consacrée à présenter l'administration des réseaux Informatique ainsi que ses différents objectifs et les protocoles qui interviennent dans cette dernière.

Quant à la deuxième partie, elle sera réservée à la gestion des réseaux Informatique dont nous allons définir ses systèmes et son architecture d'une part et d'autre part, ses fonctions et enfin la structure des systèmes de gestion des réseaux.

1.1 Présentation de l'administration des réseaux informatiques

L'administration des réseaux Informatiques évolue sans cesse et elle s'affirme aujourd'hui comme une activité clé de toute entreprise. En plus d'être constamment en fonction, ses outils d'échange de données et de partage d'information en temps réel doivent être en mesure d'offrir une confidentialité maximale et une sécurité à toute épreuve.

Un administrateur réseau est chargé de gérer l'ensemble des équipements réseaux d'une structure,tel que l'instalation du matériel, s'assurer des performances du réseau, effectuer des opérations de dépannage et configurer le matériel informatique.

L'administrateur de réseau informatique peut intervenir dans toutes les étapes de la création d'un réseau. Il planifie l'implantation ou la migration du réseau, s'assure de la disponibilité de l'équipement nécessaire à son installation, configure et rend fonctionnelle chacune de ses composantes, il supervise le fonctionnement, gère la sécurité et assure le soutien aux utilisateurs. Une part importante du travail de l'administrateur de réseaux informatiques consiste à déterminer rapidement, et avec plus de précision possible, la cause d'un problème et à appliquer une solution pertinente, tout en évaluant les conséquences de son choix sur l'ensemble du réseau.

1.1.1 Les objectifs de l'administration des réseaux Informatiques

L'objectif nodal de l'administration des réseaux est d'avoir un réseau opérationnel sans rupture de service, ce qui définit une certaine qualité de service qui se distingue sur plusieurs critères, du point de vue de l'utilisateur final, notamment la disponibilité, la performance, la fiabilité et enfin la sécurité offerte par l'opérateur.[12] L'administration des réseaux consiste à mettre en place, maintenir et organiser l'infrastructure du réseau, mais aussi :

- Assurer la sécurité des données internes au réseau.
- Gérer les systèmes des fichiers partagés et les maintenir.
- Détection et prévision des erreurs.
- Gérer l'accès au réseau (nom d'utilisateur, mot de passe, droits d'accès et permission particulière).

1.1.2 Les protocoles de l'administration des réseaux informatiques

Jusqu'à la fin des années 70, il n'y avait aucun protocole utilisé spécifiquement pour la gestion des réseaux et de leurs dispositifs, Le seul outil de gestion utilisé jusqu'à ce moment

est **ICMP** (**I**nternet **C**ontrol **M**essage **P**rotocol). Ce dernier est implanté au-dessus de **IP** (Internet Protocol) et permet d'envoyer des messages de contrôle à des routeurs et des stations.

Cependant, vers la fin des années 80, lorsque la taille de 1'Internet débuta sa croissance exponentielle une attention a été portée sur le développement d'un autre moyen de gestion, plus sophistiqué. Pour répondre à cet urgent besoin, **IAB** (**I**nternet **A**ctivities **B**oard), l'organisme qui supervise les efforts d'interconnexion des réseaux et de développement des protocoles pour la communauté **TCP/IP**, a décidé de prendre en charge la coordination des efforts pour adopter un protocole standard de gestion des réseaux. Trois protocoles ont été proposés : [3]

1.1.2.1 HEMS

(High-Level Entity Management System) il correspond à la généralisation du protocole **HMP** (**H**ost **M**onitoring **P**rotocol).

1.1.2.2 CMIS /CMIP

Le service pour échanger des informations de gestion entre stations de gestion et agents, au sein des systèmes OSI, est connu sous le nom de **CMISE** (**C**ommun **M**anagement **I**nformation **S**ervice **E**lement). CMISE est spécifié en deux parties :

- Des services fournis par chaque élément du réseau pour des buts de gestion. C'est ce qu'on appelle **CMIS** (**C**ommon **M**anagement **I**nformation **S**ervice).
- Un protocole spécifiant le format des unités de données (**PDU**) utilisées et les services qui leur sont associés, c'est le **CMIP** (**C**ommon **M**anagement **I**nformation **P**rotocol). [3]

1.1.2.3 SNMP

SNMP (Simple Network Management Protocol) est une version améliorée du protocole Simple Gateway Management Protocol **SGMP**.

Il est développé à partir des années 80 et il est devenu le standard actuel d'administration des réseaux **TCP/IP**.

Actuellement, c'est la version 03 de ce protocole qui est en cours de diffusion. Cette version est améliorée par rapport aux anciennes (version 1 et 2), en particulier sur le plan de la sécurité. [3] (***Ce protocole sera détaillé dans le chapitre 02***).

1.1.2.4 Comparaison des protocoles SNMP et CMIP/CMIS

Dans cette partie nous allons faire une comparaison entre le protocole SNMP et CMIP/CMIS en donnant un apercu général de leurs avantages et inconvenients .Cette comparaison est illustré dans le tableau [1.1].

Protocole	Avantages	Inconvénients
SNMP	-Implantation simple -Utilisation très large	-Absence de sécurité (réglée par SNMPv3) -Modèle d'information basé sur des objets décrits sous forme de variable simple -Existe uniquement sur les réseaux qui supportent IP.
CMIP/CMIS	-Sécurité développée -Modèle d'information basé sur l'approche orientée objet	-Non supporté par tous les équipements(retour, swish...)

TABLE 1.1 – Comparaison entre le protocole SNMP et CMIS/CMIP [2].

1.2 Concepts de la gestion des réseaux Informatiques

La gestion des réseaux se définit comme étant l'ensemble des moyens mis en œuvre (connaissances, techniques, méthodes, outils, ...etc.) pour superviser et exploiter des réseaux Informatiques et planifier leur évolution en respectant les contraintes de coût, de qualité et de matériel.

1.2.1 Systèmes de gestion des réseaux

Un système de gestion des réseaux est une collection d'outils de contrôle et de surveillance utilisés pour assurer le bon fonctionnement du réseau.

Généralement, il est constitué d'un ensemble de composantes logicielles et matérielles ajoutées à celles constituant le réseau en question. Les composantes logicielles utilisées pour la gestion figurent dans des ordinateurs hôtes et des processeurs de communication (tels que les ponts et les routeurs).[1]

1.2.1.1 Architecture d'un système de gestion des réseaux

Il n'existe aucune règle à appliquer pour définir l'architecture d'un système de gestion des réseaux. Cependant, en tenant compte des fonctions requises par ce système, les points suivants doivent être considérés lors de son développement :

- Le système doit posséder une interface graphique permettant de visualiser la hiérarchie du réseau et d'établir les connexions logiques entre ces niveaux hiérarchiques.
- Le système doit posséder une base de données relationnelle qui peut enregistrer n'importe quelle information requise par les applications de gestion.
- Le système doit être capable de retirer des informations à partir de tous les dispositifs connectés au réseau.
- Le système doit être facile à étendre et à personnaliser.
- Le système doit être en mesure de détecter et suivre les problèmes qui peuvent survenir.

En se basant sur ces aspects, trois approches de système de gestion des réseaux ont été établies :

- **Approche centralisée**

  Cette approche est appliquée pour les grands systèmes sur lesquels sont installées la grande majorité des applications.

  Elle consiste à faire remonter toutes les informations de gestion de chaque ressource du réseau vers un point central qui les analyse et décide de l'opération à entreprendre.

  Chaque ressource est représentée par un programme appelé agent.

  Ce dernier, communique les informations sur l'état de la ressource à la station de gestion. [8]

  Cette approche est utilisée dans l'architecture du protocole SNMPv1.

  Cette architecture est illustrée dans la figure [1.1] :

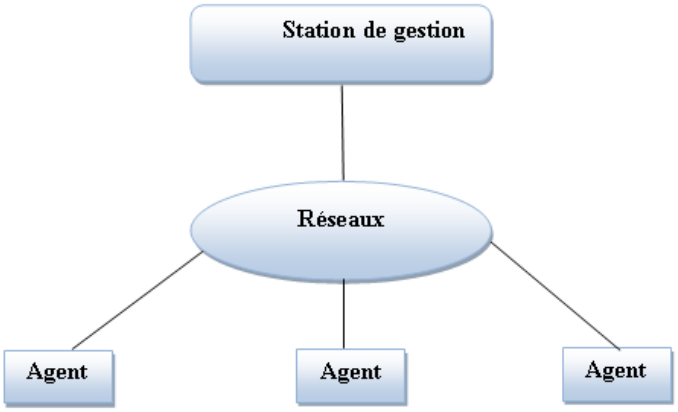

FIGURE 1.1 – Approche centralisée. [8]

- **Approche distribuée**

  Elle regroupe plusieurs systèmes de gestion qui tournent simultanément sur le réseau. Cette approche est constituée de plusieurs gestionnaires des domaines indépendants qui communiquent entre eux pour s'échanger des informationx sur l'état du réseau.

  Chacun des gestionnaires est responsable de son propre domaine.

  Cette architecture permet d'augmenter la fiabilité et la performance des systèmes de gestion des réseaux. Un exemple d'architecture distribuée est le réseau Internet.[8]

  La figure [1.2] illustre le fonctionnement de cette approche :

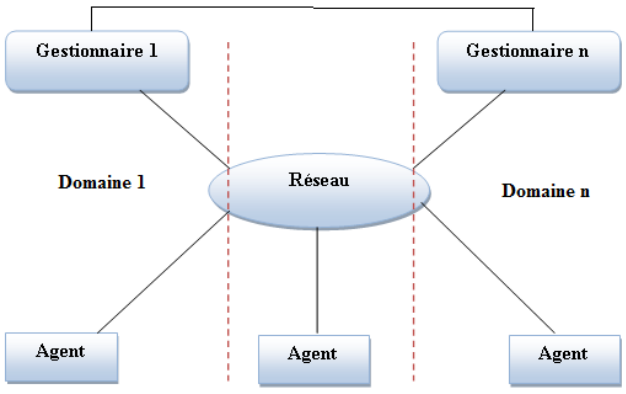

FIGURE 1.2 – Approche distribuée. [8]

- **Approche hiérarchique**

  C'est la combinaison des deux architectures précédentes.

  Le système central correspond à la racine de la hiérarchie. Il permet de sauvegarder quelques données dans l'unité centrale de stockage des données et contrôle l'accès aux

différentes parties du réseau.

Cette approche utilise le concept gestionnaire des gestionnaires. Dans ce concept, chaque gestionnaire est responsable de la gestion de son domaine, lequel est constitué d'un ensemble d'agents. Les gestionnaires de domaine ne communiquent pas directement entre eux ; ils communiquent uniquement avec le gestionnaire central.

Cette approche est utilisée dans l'architecture des protocoles CMIP et SNMPv2. [8]

La figure [1.3] illustre le fonctionnement de l'architecture :

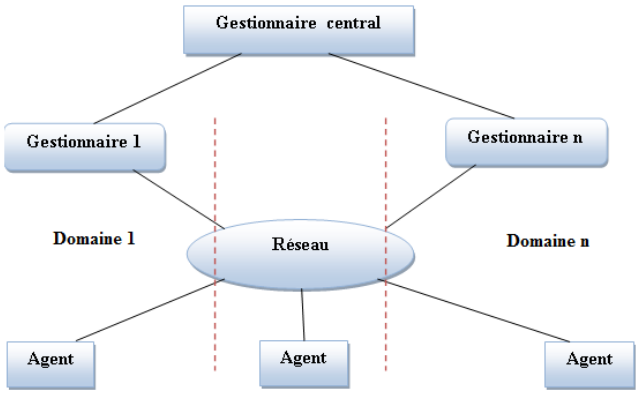

FIGURE 1.3 – Approche hiérarchique.[8]

1.2.2 Fonction de gestion des réseaux

Les applications de gestion des réseaux consistent à planifier, à contrôler et à anticiper sur l'évolution probable d'un réseau, d'un protocole ou sur la possibilité d'une anomalie. Elles assurent également la correction de ces anomalies.

Le processus de gestion des réseaux inclut généralement la collecte des données, faite automatiquement ou via l'analyste responsable du réseau, le traitement de ces données ensuite, leur présentation.

Pour répondre à ces besoins, l'ISO (**I**nternational **O**rganization for **S**tandardisation) a défini cinq domaines fonctionnels de la gestion des réseaux [15] :

1.2.2.1 Gestion de configuration

La gestion des configurations effectue un suivi des différentes configurations des éléments présents sur le réseau.

Elle stocke dans une base de données les versions des systèmes d'exploitation et des logiciels installés sur chaque machine du parc réseau.

La gestion des configurations permet donc une identification et un contrôle des systèmes ouverts ; Elle collecte et fournit des informations sur les différents systèmes du réseau.

1.2.2.2 Gestion de performance

La gestion des performances analyse de manière continue les performances du réseau afin de le maintenir dans un état de performance acceptable.

Elle permet d'évaluer les performances des ressources du système et leur efficacité. Les performances d'un réseau sont évaluées à partir de quatre paramètres : le temps de réponse, le débit, le taux d'erreur par bit et la disponibilité.

1.2.2.3 Gestion des pannes

L'objectif principal de la gestion des pannes est de détecter, d'isoler et de corriger à la fois les anomalies qui surviennent sur le réseau.

Elle essaie d'isoler le plus précisément le problème en effectuant divers tests, quand cela est possible, elle règle elle-même automatiquement l'anomalie. Sinon, elle alerte les personnes concernées par le type de problème afin de solliciter leur intervention.

La gestion des pannes garde dans une base de données l'ensemble des problèmes survenus ainsi que leur solution.

1.2.2.4 Gestion de sécurité

Elle consiste à contrôler l'accès aux ressources et à assurer la confidentialité et l'intégrité de l'information transmise sur le réseau ; Elle assure également l'authentification de l'émetteur.

1.2.2.5 Gestion de la comptabilité

Le but de la gestion de l'information comptable est de mesurer les paramètres d'utilisation du réseau afin de gérer convenablement l'exploitation des ressources disponibles.

Ces mesures peuvent être utilisées pour évaluer l'utilisation des ressources ou pour facturer l'exploitation de ces ressources.

1.2.3 Structure des systèmes de gestion des réseaux

Pour mieux décrire l'architecture de ces systèmes de gestion, nous décomposons chaque architecture en trois modèles distincts : le modèle informationnel, le modèle de communication

et le modèle organisationnel . Ces modèles englobent les parties essentielles d'un système de gestion des réseaux.[7]

1.2.3.1 Le model informationnel

Ce modèle fournit une vue du réseau par les données et structure de l'information de gestion, ces informations définissent les besoins de gestion des ressources matérielles et logicielles existantes sur le réseau. Elles sont stockées dans une base de données appelée Management Information Base (**MIB**).

Au niveau de la normalisation, l'**ISO** définit ces informations de gestion comme des objets stockés dans une **MIB**. Ces objets sont définis suivant le concept orienté objet.

Par contre, l'**IETF** représente ces informations de gestion à travers des variables stockées dans une base de données virtuelle. Le modèle informationnel constitue la base sur laquelle reposent les deux modèles qui suivent.

1.2.3.2 Le model organisationnel

Ce modèle définit les entités de gestion qui échangent des informations de contrôle en utilisant le modèle de communication. Il repose essentiellement 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 MIB sur laquelle le gestionnaire peut agir.

Ce concept de gestionnaire/agent est repris par tous les organismes de normalisation.

1.2.3.3 Le model de communication

Quant à ce modèle, il décrit comment l'information de contrôle est acheminée entre les entités de gestion. Il fournit les moyens de recueil des informations élémentaires ou statistiques auprès des agents représentant les ressources.

Pour cela, l'**ISO** a défini le protocole **CMIP** et l'**IETF** a défini le protocole **SNMP** qui sera aborder dans le prochain chapitre.

Conclusion

Dans ce chapitre, nous avons abordé les aspects architecturaux des applications et des systèmes de gestion des réseaux informatiques, dont nous avons exposé les points et les étapes essentielles de la gestion des réseaux.

Dans le chapitre suivant nous allons faire une étude de protocole SNMP, dont nous présenterons ses différentes versions, ainsi que ses fonctionnalités.

2

Etude du protocole SNMP

Introduction

Dans ce chapitre nous allons donner un aperçu sur le protocole **SNMP**, en premier lieu nous présenterons ses aspects de base, ensuite nous allons aborder la notion du **MIB** utilisés par les systèmes de gestion des réseaux. Enfin, nous allons donner une esquisse de la version 3 du même protocole, dont nous allons mentionner juste les améliorations apportées sur ce dernier.

2.1 Présentation du protocole SNMP

SNMP (**S**imple **N**etwork **M**anagement **P**rotocol) est un protocole simple de gestion de réseau développé par un groupe de travail de l'**IETF** (**I**nternet **E**ngineering **T**ask **F**orce) dans le cadre de la définition d'un système de gestion pour les réseaux utilisant les protocoles **TCP/IP**.

Etant un protocole Internet, il est compatible avec toutes les plateformes hétérogènes et est installé sur la plupart des équipements réseaux tels que les routeurs, commutateurs,... etc. **SNMP** a été approuvé par l'**IAB** (**I**nternet **A**ctivities **B**oard), responsable des spécifications de TCP/IP. Plusieurs documents définissent ce standard, parmi lesquels :[2]

- RFC 1155 SMI (Structure of Management Information) ;
- RFC 1156 MIB (Management Information Base) ;
- RFC 1157 SNMP Protocol ;
- RFC 1158 MIB II (Management Information Base II) ;

2.2 L'architecture du protocole SNMP

La figure [2.1] illustre l'architecture du protocole SNMP en indiquant les liens existants entre le gestionnaire, l'agent SNMP et les objets gérés par celui-ci.

Dans ce qui suit, nous détaillerons les éléments illustrés dans cette figure.

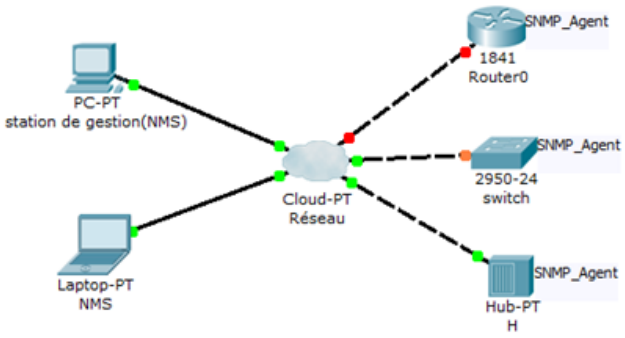

FIGURE 2.1 – L'architecture du protocole SNMP .

- **Une station de gestion NMS** (***N**etwork **M**anagement **S**tation)* : est la station qui exécute les applications dans le but de contrôler les éléments des réseaux. Physiquement, la station NMS est un poste de travail avec un processeur rapide et qui nécessite beaucoup de mémoires et d'espace sur le disque.
- **Les éléments du réseau** : Ce sont les éléments à gérer sur le réseau. Cela va d'une station NMS à un commutateur, routeurs, concentrateurs,...etc.
- **Agent SNMP** : est installé dans la majorité des éléments du réseau que nous avons cités au dessus. Cet agent connaît les paramètres du périphérique sur lequel il s'exécute. Son rôle est de répondre aux requêtes de la station (**NMS**).
- **Les tables MIB** : (***M**anagement **I**nformation **B**ase*) est une collection d'objets résidant dans une base d'information virtuelle. Des collections d'objets reliés sont définies dans des modules MIB spécifiques.

2.3 Principe de fonctionnement du protocole SNMP

Le fonctionnement de **SNMP** est asymétrique ; il est constitué d'un ensemble de requêtes, des réponses et d'un nombre limité d'alertes. La station de gestion envoie des requêtes à l'agent, lequel retourne des réponses. Lorsqu'un événement anormal surgit sur l'élément de réseau, l'agent envoie une alerte (trap) à la station de gestion de réseau. [6]

La figure [2.2] illustre le principe de fonctionnement du SNMP :

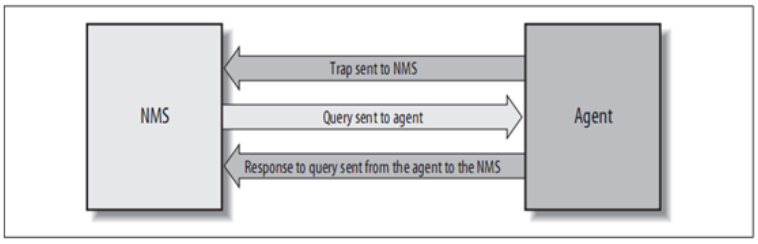

FIGURE 2.2 – fonctionnement de SNMP [6].

2.4 Structure de l'information de gestion

La syntaxe des informations de gestion, intitulée **SMI** (Structure and Identification of Management Information for TCP/IP Based Internet), définit comment chaque élément

d'information, concernant les périphériques gérés et les agents, est représenté dans la base d'information de gestion.

2.4.1 Structure du MIB

La plupart des matériels et des logiciels réseaux possèdent une base de données stockée dans le matériel ou dans le logiciel appelé **MIB**.

La **MIB** est une base de données d'administration du réseau. Elle définit et décrit tous les éléments d'informations nécessaires à l'administrateur du réseau autant sur le plan technique que sur le plan administratif.

Les objets de la MIB sont hiérarchisés en fonction de leur nature fonctionnelle. Cette hiérarchie peut se visualiser sous la forme d'une arborescence appelé Arbre d'informations d'administration (**MIT** : ***M**anagement **I**nformation **T**ree*).[6] comme est illustré à la figure [2.3] :

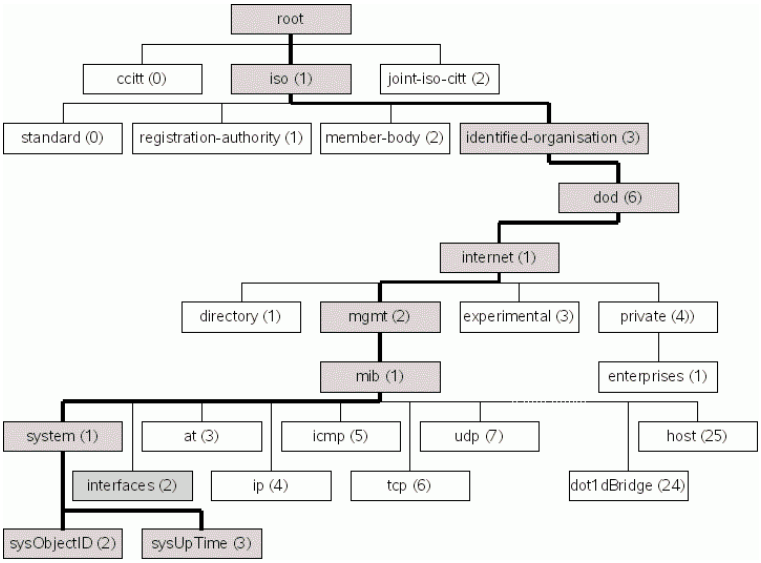

FIGURE 2.3 – Structure de la table MIB.

A la racine, sont liés par filiation trois nœuds : le **CCITT**, l'**ISO**, l'union **ISO-CCITT**.

La branche menant aux informations relatives à l'objet SNMP commence au sous-arbre ISO lequel se divise en quatre branches : Standard, Registration-Authority, Member-Body et Identified-Organization.

De la branche Identified-Organization, partent six nouvelles branches dont celle du **DoD** (***D**épartement **of D**éfense*) qui a alloué son premier nœud à l'**IAB**. De l'**IAB** partent quatre autres branches : Directory, Management, Expérimental et Private.

La branche Management (Administration) contient les éléments (dits objets) définis dans les documents de l'**IAB** tels que les **RFC**.

La branche PRIVATE contient le sous-arbre Enterprise, destiné aux entreprises privées et organisations qui souhaitent développer des objets propres à leurs équipements (MIB privées). Par ailleurs, ces entreprises peuvent apporter des extensions à la MIB standard.

La branche Expérimental est utilisée pour décrire les objets situés dans un état non définitif.

L'identifiant d'un objet (Object Identifier) se présente sous la forme d'une suite d'entiers décrivant sa position dans l'arbre de la racine jusqu'au nœud correspondant.

2.4.2 Structure des objets MIB

Les objets, au sein d'un MIB, sont définis en utilisant une syntaxe appelée **ASN.1** (***A****bstract* ***S****yntax* ***N****otation One*) .Ce dernier est un langage formel qui a été développé et standardisé par la **ITU-T** et L'**ISO**. Il est utilisé pour :

- Définir une syntaxe abstraite pour les applications ;
- Définir la structure des unités de données (Protocol Data Unit, PDU) des couches application et présentation ;

Le tableau [2.1] illustre les types utilisés pour définir les objets au sein de la MIB.

L'ensemble de ces types de données est divisé en trois catégories :

- **Types universels** : cette catégorie regroupe les types suivants : INTEGER, OCTET STFUNG et OBJECT IDENTIFIER.
- **Types application** : regroupe les types suivants : Integer32, Unsigned32, Gauge32, Counter32, Counter64, TimeTicks, IpAddress et Opaque.
- **Pseudo-types** : cela inclut uniquement le type BITS.

2.4.3 Définition des objets MIB

La définition d'un objet au sein d'une MIB est effectuée à base de macros. La macro utilisée pour définir un objet MIB est nommée **OBJECT-TYPE**.Cette dernière est reconnue dans la version de SMI. [3]

Les différents champs utilisés par OBJECT-TYPE sont les suivants :

Nom	Descriptions
Integer32 ; Unsignecl32	Spécifier des valeurs pouvant être négatives ou positives
Gauge32	Est utilisé pour spécifier les types de données dont les valeurs ne pouvant pas dépasser des bornes prédéfinies.
Counte32/ Counte64	-utiliser pour compter la production d'un événement -mesurer un flux de données.
Time Ticks	Permet de spécifier des unités de données exprimées en seconde.
OCTET STRING	est utilisé pour spécifier des octets pouvant contenir des informations binaires ou textuelles.
Opaque	permet de ne spécifier que des octets de valeur binaire.
OBJECT IDENTIFIER	est utilisé pour identifier un objet
IpAddress	Spécifier les adresses réseaux.
BITS	Sert à regrouper un ensemble de bits prénommés.

TABLE 2.1 – les types utilisés par SNMP.

- ***SYNTAX*** : il permet de spécifier la syntaxe d'un objet. Cette syntaxe est construite en utilisant les types universels et les types application.
- ***MAX-ACCESS*** : il définit le mode d'accès permis à une instance d'un objet. Les valeurs possibles sont : read-only, read-write, read-creute, not-accessible et accessible- for-notih.
- ***STATUT*** : désigne la validité de la définition de l'objet. Les valeurs possibles de ce champ sont current (valide), deprecoted (remplacée par une autre) et obsolete (non valide et ne peut plus être appliquée).
- ***DESCRIPTION*** : il correspond à une description textuelle (chaîne de caractères) de la sémantique du type d'objets en question.
- ***UNITS*** : contient une définition textuelle des unités associées à un objet (par exemple : seconde pour le cas du temps).
- ***REFERENCE*** :contient une référence textuelle à un objet défini dans d'autres modules MIB.
- ***INDEX*** : définit comment les rangées d'une table sont indexées. Ce champ illustre de façon ordonnée les différents champs qui entrent dans la composition de l'index d'une table. Il n'y a aucune restriction sur le nombre d'index. Typiquement, ils correspondent à des objets colonnes de la table.
- ***AUGMENTS*** : permet d'étendre le nombre de colonnes d'une table sans toucher à sa

structure (sans redéfinir la table). AUGMENTS reçoit comme paramètre l'identificateur d'une rangée d'une autre table. Cette dernière est appelée table de base par contre, la table utilisant le champ AUGMENTS est appelée table d'augmentation. Une table de base peut être étendue par plusieurs tables d'augmentation. Cet aspect est semblable à la notion d'héritage.

- ***DEFVAL*** : spécifie la valeur à affecter à une instance d'un objet colonne lors de sa création. Aucune valeur initiale n'est attribuée aux objets correspondant aux index d'une table, ni à un objet scalaire. Seuls les objets colonnes accessibles en mode read-create peuvent avoir un champ DEFVAL.

Les champs SYNTAX, ACCESS (ou MAX-ACCESS) et STATUS (INDEX ou AUGMENTS pour les tables) sont obligatoires lors de la définition d'un objet. Par contre, les champs UNITS, REFERENCE, DESCRIPTION et DEFVAL sont optionnels.

2.4.4 Définition des tables

La macro OBJECT-TYPE est utilisée pour définir une table d'objets.

Une table SNMP est composée d'un ensemble de rangées et de colonnes. L'argument du champ SYNTAX pour une table doit être SEQUENCE OF "séquence".

Enfin, la valeur du champ ACCESS ou du champ MAX-ACCESS d'une table doit être not-accessible (vu qu'une table n'est pas directement accessible avec les opérations de SNMP).

Suite à la définition d'une table, vient la définition de ses rangées. Le constructeur OBJECT-TYPE est aussi utilisé dans ce but. L'argument du champ SYNTAX d'une rangée doit être un identificateur pour une séquence (l'identificateur de la rangée). La valeur des champs ACCESS ou MAX-ACCESS doit être aussi not-accessible.

La valeur du champ STATUS d'une rangée donnée doit être la même que celle de la table. Le champ INDEX ou AUGMENTS spécifie comment les instances des objets colonnes de la table sont identifiées. SM1 mentionne que la valeur OID attribuée à une rangée doit être la même que celle de la table dont elle fait partie, tout en ajoutant la valeur 1 à la fin de cette même valeur OID.

Par exemple : si la valeur OID d'une table nommée printerTable est x, alors la valeur OID de sa rangée (printerEntry) sera x.1. [3]

Voici un exemple de définition d'une table :

IFTable OBJECT-TYPE
SUNTAX sequence of entry
MAX ACCESS not accessible
STATUS current

2.5 Versions de SNMP

SNMP évolue et se décline actuellement en trois versions. Les versions 1 et 2 sont proches et garantissent la compatibilité descendante. Nous décrivons ici les particularités propres à chaque version ; [14]

- **SNMPv1 (complet)** : Ceci est la première version du protocole, tel que définie dans le RFC 1157. Ce document remplace les documents plus ancien comme RFC 1067 et RFC 1098. On dit que la sécurité de cette version est triviale, car la seule vérification qui est faite est basée sur la chaîne de caractères « communauté ».
- **SNMPsec (historique)** : Cette version ajoute de la sécurité au protocole SNMPv1, elle est définie par RFC 1351, RFC 1352 et RFC 1353. La sécurité est basée sur des groupes. Très peu ou aucun manufacturier n'a utilisé cette version qui est maintenant largement oubliée.
- **SNMPv2p (historique)** : Beaucoup de travaux ont été élaborés pour faire une mise à jour de SNMPv1. Ces travaux ne portaient pas seulement sur la sécurité. Le résultat est une mise à jour des opérations du protocole, des nouvelles opérations, des nouveaux types de données. La sécurité est basée sur les groupes de SNMPsec. Cette version est décrite par RFC 1441, RFC 1445, RFC 1446, RFC 1448 et RFC 1449.
- **SNMPv2c (expérimental)** : Cette version du protocole est appelée « community string-based SNMPv2 ». Ceci est une amélioration des opérations du protocole et des types d'opérations de SNMPv2p et utilise la sécurité par chaîne de caractères « communauté » de SNMPv1. Cette version est définie par RFC 1901, RFC 1905 et RFC 1906.
- **SNMPv2u (expérimental)** : Cette version du protocole utilise les opérations, les types de données de SNMPv2c et la sécurité basée sur les usagers. Cette version est décrite par RFC 1905, RFC 1906, RFC 1909 et RFC 1910.
- **SNMPv2* (expérimental)** : Cette version combine les meilleures parties de SNMPv2p et SNMPv2u.

- **SNMPv3** : Cette version comprend une combinaison de la sécurité basée sur les usagers et les types et les opérations de SNMPv2p, avec en plus la capacité pour les « proxy ». La sécurité est basée sur ce qui se trouve dans SNMPv2u et SNMPv2*. Le standard SNMPv3 sera détaillé dans la section protocole SNMPv3

2.6 Opérations de SNMP

Le protocole SNMP est un protocole sans connexion, qui utilise principalement le protocole UDP (User Datagram Protocol). Un système SNMP supporte trois types de requêtes : GET, SET et TRAP.[2]

La figure [2.4] illustre les types d'opérations supportées par SNMP.

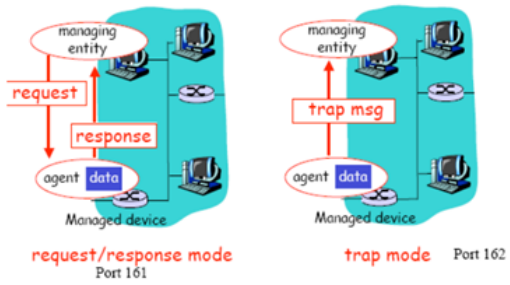

FIGURE 2.4 – les Operations de SNMP. [5]

Les opérations applicables avec SNMP sont les suivantes[2] :

- **GetRequest** : Cette requête permet aux stations de gestion (manager) d'interroger un agent sur les valeurs d'un objet d'une MIB.
- **GetNextRequest** : Quant à celle-ci, elle permet aux stations de gestion de recevoir le contenu de l'instance qui suit l'objet nommé (passé en paramètre) dans la MIB. Cette commande permet en particulier aux stations de gestion de balayer les tables des MIB. Elle permet également d'accéder à plusieurs variables simultanément.
- **GetResponse** : est le message retourné par les entités interrogées (agents) en réponse aux commandes de type GET REQUEST, GET NEXT REQUEST et SET REQUEST.[2]
- **GetBulkRequest** : (version 2 et version 3) Cette requête est une amélioration du SNMP, elle permet aux stations de gestion d'interroger l'agent sur un ensemble de variable dans la MIB.

- **SetRequest** : Cette requête permet aux stations de gestion de modifier une valeur de la MIB ou d'une variable et de lancer des périphériques. Elle permet par exemple à un manager de mettre à jour une table de routage.
- **Les alarmes TRAP** : Lorsqu'un périphérique entre dans un état anormal, l'agent SNMP prévient le gestionnaire SNMP par le biais d'une Trap SNMP. Les messages Trap peuvent être cold-start (démarrage à froid), warm-start (démarrage à chaud), réinitialisation de l'agent SNMP, authentification failure (échec d'authentification, lorsqu'un nom de communauté incorrect est spécifié dans une requête), loss-of -EGP neighbour (perte de voisin EGP),etc.
- **InformRequest** : (version 2 et version 3) cette requête permet à un manager d'envoyer une information non sollicitée à un autre manager. Il peut, par exemple, signaler un débit excessif sur une ligne de communication.

2.7 Les communautés SNMP

SNMP v1 et v2 propose des options de sécurité assez basiques, elles sont basées sur les communautés. Une communauté est un groupe qui possède des accès communs à un ensemble d'objets de la MIB, un peu comme un groupe d'utilisateurs. Une communauté est une relation entre un agent SNMP et un ensemble de managers SNMP.

Les noms sont définis sur chaque agent et donc, la communauté "local" de l'agent1 peut avoir des accès différents à la communauté « local » de l'agent2. Il suffit juste de connaitre le nom de la communauté pour avoir les droits qui lui sont liés, le nom de la communauté est utilisé comme un mot de passe.

En effet, le nom de la communauté transite en clair sur le réseau. Un message SNMP contient la communauté et d'ou n'importe qui pourrait récupérer le nom d'une communauté en analysant le trafic réseau. Il est donc dangereux de donner un accès en écriture à une communauté car n'importe qui pourrait y avoir accès. [5]

2.8 Identification des instances

Un schéma est utilisé par SNMP pour identifier les entités faisant partie du système de gestion. Un identificateur dans ce schéma est appelé identificateur d'objet (Object Identifier, OID) et l'identité d'un objet est déterminé par la valeur de son OID. L'affectation d'une valeur OID à une entité est appelée enregistrement.

Une fois que l'enregistrement a été effectué, aucune autre entité ne peut être inscrite avec la même valeur d'OID.

Par définition, L'identificateur d'un objet est une séquence d'entiers séparés par un point(.). On dispose alors d'une structure hiérarchique représentée sous forme d'un arbre (voir la Figure [2.5]). La racine n'est pas numérotée ; chaque nœud de l'arbre décrit un identificateur d'objet et ces feuilles correspondent aux instances de l'objet.[5]

Il ya deux techniques pour identifier une instance d'objet avec SNMP :

- **Serial-access** : Basé sur l'ordre lexicographique des objets, utile pour accéder séquentiellement aux objets (GetNext Request).
- **Random-access** : Accès direct à l'instance d'un objet.

  L'instance d'un objet scalaire d'une ligne particulière d'un tableau est la concaténation de :

  – L'OBJECT IDENTIFIER du tableau
  – Le suffixe qui identifie l'objet ligne du tableau
  – Le suffixe qui identifie l'objet scalaire dans la ligne
  – Un ensemble de valeurs de l'INDEX de l'objet

Exemple :

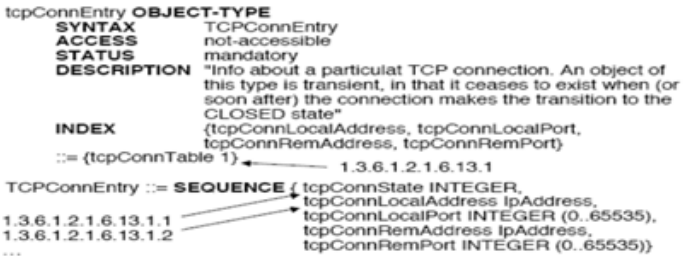

FIGURE 2.5 – exemple d'une instance d'objet

2.9 Description du protocole SNMP

Dans cette section, nous allons présenter le format des messages SNMP.

2.9.1 Le format des messages

Les informations échangées entre le manager et l'agent sont sous forme d'un message ; chaque message intègre le numéro de version de SNMP, le nom de communauté et un PDU.[13]

la figure [2.6] montre le format des messages :

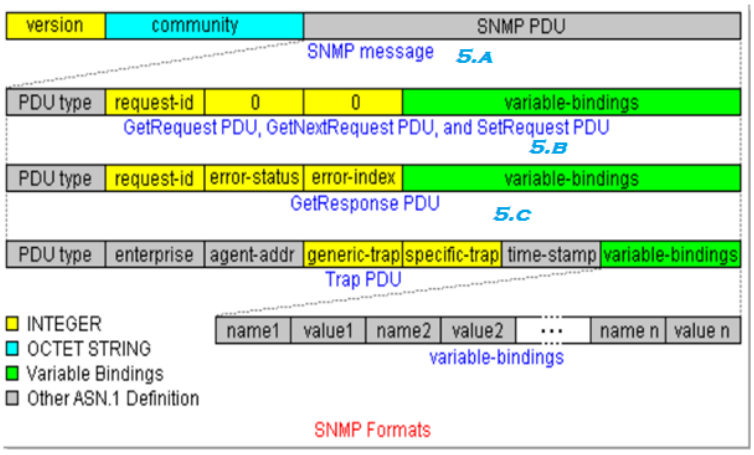

FIGURE 2.6 – Format des messages SNMP [13]

Description des champs d'un message SNMP sont démantrés dans le tableau [2.2] :

Version	Version de SNMP
Community	Nom de la communauté (agit comme un mot de passe).
Request-id	Utilisé pour différencier les messages.
Error-status	Utilisé pour signaler une erreur (0 si pas d'erreur).
Error-index	Indique la sous-catégorie d'erreur.
Variablebindiings	Nom des variables avec leurs valeurs.
Enterprise	Type de l'objet générant l'alarme.
Agent-addr	Adresse de l'émetteur de l'alarme.
Generic-trap	Identificateur de l'alarme.
Specific-trap	Identificateur d'alarme spécifique.
Time-stamp	Temps écoulé depuis la dernière réinitialisation de l'entité.

TABLE 2.2 – description des champs du message SNMP.

2.9.2 Envoi d'un message

La transmission des PDU se fait entre entités (que ce soit d'une station vers un agent ou l'inverse). [13]

Les actions suivantes sont effectuées lors de la transmission d'un PDU :

◦ Utilise ASN.1 pour créer un PDU.

- Ce PDU est émis à un service d'authentification, avec des adresses sources et destination ainsi que le nom de la communauté, le service d'authentification va alors exécuter leurs opérations et les opérations de cryptage.
- Le message est construit à partir du PDU avec l'ajout du nom de la communauté et la version de SNMP.
- Ce nouveau message est ensuite codé puis passé au service de transport, qui va le livrer à l'agent SNMP spécifié.

2.9.3 Réception des messages

Lors de la réception d'un message, l'entité SNMP passe par les étapes suivantes [13] :

- Le message est reçu et se voit opérer une vérification syntaxique. Si le message est défectueux, il est ignoré.
- Le numéro de version est vérifié, s'il n'est pas conforme, le message est ignoré.
- Le nom d'utilisateur, le PDU, l'adresse de source et de destination au niveau transport, sont émis à un service d'authentification.

  - Si l'authentification échoue, le service prévient l'entité transport de SNMP, laquelle envoie une alarme et ignore le message.

  - Si l'authentification réussit, le service renvoie un PDU de la forme d'un objet ASN.1 qui conforme à la norme RFC 1157.
- La syntaxe du message retourné sera vérifiée. Ce message est ignoré dans le cas ou la vérification est échouée.

2.10 Le protocole SNMP version 3

Les deux versions de SNMP ont rencontré plusieurs failles en ce qui concerne la sécurité ; c'est pourquoi, il a fallu de mettre en œuvre un mécanisme permettant de sécuriser les transactions, et ainsi offrir une gestion plus sûre à l'administrateur.

SNMP v3 offre de nouvelles capacités d'ouverture et d'interopérabilité de gestion. Les spécifications de SNMPv3 sont basées sur une architecture modulaire ; une entité SNMP (gestionnaire ou agent) est composée d'un moteur SNMP auquel on associe une ou plusieurs applications.

2.10.1 Les objectifs de protocole SNMPv3

La nouvelle version du protocole SNMP vise essentiellement à assurer la sécurité des transactions. Pour ce faire, nous citons quelques objectifs de cette architecture :

- Répondre aux besoins de la sécurisation des commandes de mise à jour des agents (Commande SET), qui était un point inefficace dans les versions SNMPv1 et SNMPv2c.
- Pouvoir évoluer une partie de SNMP, sans avoir à renouveler toute l'architecture.
- L'utilisation du matériel existant. Se baser sur les études des versions SNMPv2u et SNMPv2*.

2.10.2 Architecture SNMPv3 d'une plate-forme de gestion

L'entité NMS comporte un moteur SNMP et plusieurs applications.

Le moteur SNMP se décompose en trois modules :

- Transporteur (transport mapping) ;
- Module traitement des messages (message Processing) ;
- Le système de sécurité (Security Subsystem).

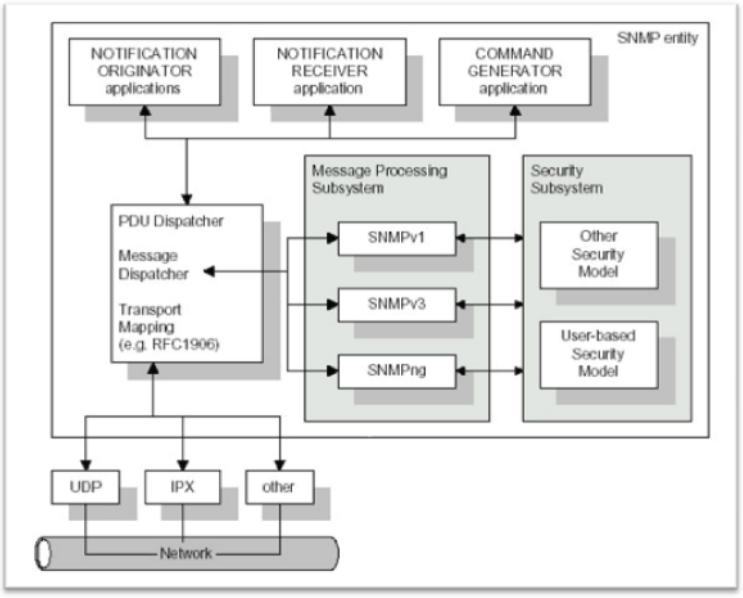

FIGURE 2.7 – Architecture SNMPv3 d'une plate-forme de gestion

2.10.3 Architecture SNMPv3 d'un agent

Dans un agent SNMPv3, on retrouve le module « access control » en outre, par rapport à l'entité NMS.

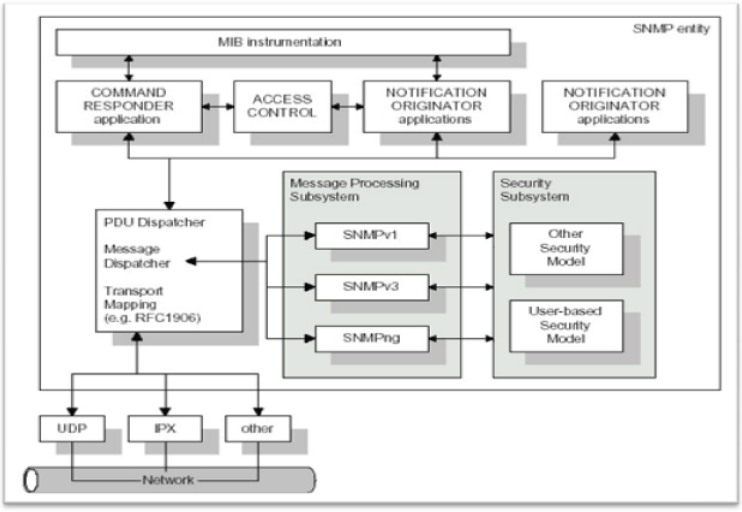

FIGURE 2.8 – Architecture SNMPv3 d'un agent

2.10.3.1 Description des champs de chaque entité

Nous allons détailler ici chaque module de l'entité NMS. [3]

- **Transporteur** (***transport mapping***) son rôle est :

  -Envoyer et recevoir des messages ;

  -Déterminer la version du message reçu afin de la faire passer vers la procédure de traitement des messages correspondante ;

  -Offrir une interface pour la transmission des PDUs d'une entité vers une autre entité SNMP distante.

- **Traitement des messages** :Le rôle de la procédure de traitement des messages est de préparer les messages à émettre et d'extraire les informations d'un message reçu. Une procédure de traitement des messages peut contenir plusieurs modules de traitement. comme est illustré dans la figure [2.9].

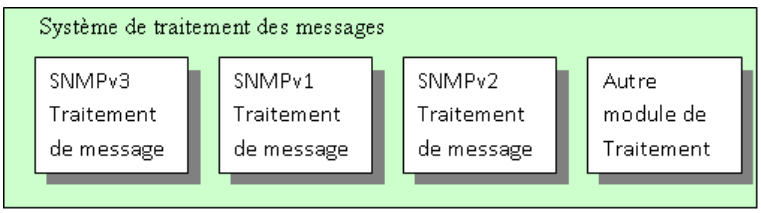

FIGURE 2.9 – les modules de traitement des messages.

Ce numéro de version permet à un moteur SNMP de déterminer à quel module de traitement ce message est destiné. Chaque module de traitement prend en charge une version de SNMP.

- **Système de sécurité** :Le système de sécurité offre les services de sécurité, comme l'authentification et la personnalisation d'un message. Ce système peut contenir plusieurs modules de sécurité comme il est illustré dans la figure [2.10]

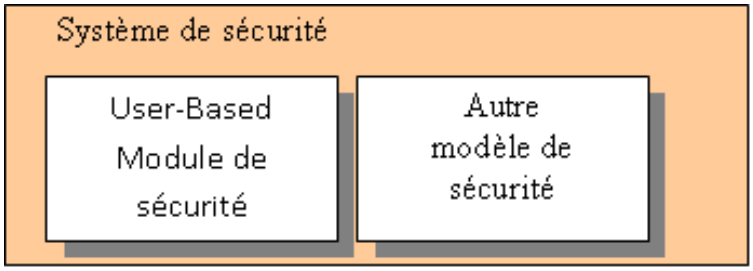

FIGURE 2.10 – les modules de sécurité.

Le module de sécurité assure les fonctions l'authentification, le chiffrement des messages et la vérification du temps.

- **Le contrôle d'accès** :Le contrôle d'accès fournit une fonction qui permet de définir les autorisations.
- **Les applications** :Les applications SNMP peuvent être regroupées en cinq types :
  - Des applications d'initiation de requête Get, GetNext, GetBulk et Set, qui sont appelées 'Command Generator'.
  - Des applications qui répondent aux requêtes Get, GetNext, GetBulk et Set, sont appelées 'Command Responder'.
  - Des applications qui génèrent des notifications, appelées 'Notification Originators'.
  - Des applications qui reçoivent des notifications, appelées 'Notification Receivers'.
  - Des applications qui transitent des requêtes SNMP Get, GetNext, GetBulk et Set, ou des notifications, appelées 'Proxy Forwarder'.

2.10.4 Format des paquets SNMPv3

Le paquet SNMPv3 suit le format suivant :

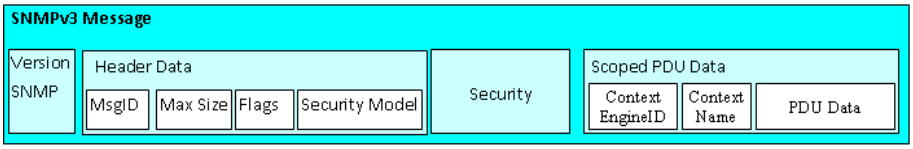

FIGURE 2.11 – Description du paquet SNMPv3.

- **MsgID** : Identificateur de message, utilisé pour repérer les requêtes et les réponses.
- **Max Size** : Taille maximale du paquet réponse, ce champ indique à la station génératrice de la réponse de ne pas dépasser cette taille qui est limité par les capacités des mémoires tampons du moteur générateur de la requête.
- **Flags** : Drapeaux, indiquant si une réponse est attendue et si un modèle de sécurité a été utilisé.
- **Modèle de sécurité** : permet de redresser le paquet vers le module de sécurité destiné.
- **Information de sécurité** : comme les clés publiques et les arrangements entre les protocoles de sécurité.
- **Identifier le contexte** : contient les informations sur le port de routeur.
- **PDU DATA** : prend en charge les valeurs demandées ou bien des réponses à des demandes.

2.10.5 La sécurité dans SNMPv3

La sécurité intégrée dans la version 3 est de type « User Based Sécurity ». Elle se repose sur l'utilisation du concept utilisateur, qui est identifié par un nom d'utilisateur « UserName » ; Chaque nom d'utilisateur est allié à un ensemble d'informations de sécurité.

2.10.5.1 Le fonctionnement de sécurité dans SNMPv3

Cette nouvelle version du protocole SNMP vise essentiellement à inclure la sécurité des transactions.

Cette sécurité est basée sur deux concepts :

- **USM** : (***U**ser-based **S**ecurity **M**odel*).
- **VACM** : (***V**iew-based **A**ccess **C**ontrol **M**odel*).

- **User Sscurity Module (USM)**

  Trois mécanismes sont utilisés. Chacun de ces mécanismes a pour but d'empêcher un type d'attaque.

  - **Authentification et intégrité** : l'intégrité a pour rôle d'assurer que le paquet reste inchangé pendant la transmission tandis que l'authentification garantit l'identité de l'émetteur.
  - **L'estampillage du temps** : Empêche la réutilisation d'un paquet SNMPv3 valide déjà transmis par un manager ; si une requête est transmise, les mécanismes d'authentification et de cryptage n'empêche pas quelqu'un de saisir un paquet SNMPv3 valide du réseau et de tenter de le réutiliser plus tard, sans modification.
  - **L'encryptions** : Empêche quiconque de lire les informations de gestions contenues dans un paquet SNMPv3.

- **View Access Control Model (VACM)**

  Permet de définir des vues sur la MIB, vues sur lesquelles les opérations SNMP pourront être limitées.

  Par exemple, il sera possible d'accéder à des tables de configuration en lecture mais pas en écriture, ou encore il ne sera pas possible du tout d'accéder, même en lecture à certaines parties de la MIB.

Conclusion

Les systèmes de gestion des réseaux utilisent des protocoles standards qui permettent de fournir des services élémentaires servant à gérer à distance les ressources de réseaux. SNMP est le protocole le plus répandu dans ce domaine.

Il fournit un ensemble d'opérations permettant de récupérer la valeur d'un objet en interrogeant la table MIB.

Dans ce chapitre nous avons, en premier lieu, détaillé les notions de base du protocole, ensuite nous avons expliqué la structure d'information.

Enfin, nous avons donné un aperçu sur les améliorations apportées à la version 3 du protocole SNMP.

dans le prochaine chapitre nous allons aborder la phase d'analyse et conception.

3

Analyse des besoins et conception

Introduction

La conception de logiciel est un art qui nécessite de l'expérience et elle consiste à traduire les besoins en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation.

Dans ce présent chapitre, nous essayons d'étendre la représentation de notre projet et le processus de modélisation (UML), nous allons détailler la phase d'analyse des besoins d'une part dont nous allons exposer les besoins fonctionnels et non fonctionnels de notre projet, et nous allons décrire les différents acteurs du système et ses cas d'utilisation.

D'autre part, nous allons concevoir les classes et les attributs et nous allons détailler le diagramme de classe.

3.1 Présentation du projet

Un réseau local ne doit pas être considéré uniquement comme un ensemble d'équipements reliés entre eux une fois pour toutes. Il doit être entretenu et surveillé. Son administrateur doit connaître les performances de chaque section et être en mesure de détecter facilement et rapidement les pannes.

Une bonne administration permettra de mieux résister aux intrusions, seuls les utilisateurs autorisés accèdent aux ressources et les menaces sont vite détectées. Pour ce faire, l'IETF a proposé le protocole **SNMP**.

Notre projet de fin d'étude répond à des besoins sentis par le personnel administratif et les employés de l'entreprise d'accueil ***SONATRACH***, pour la mise en place d'un système de gestion réseau qui se base sur le protocole **SNMP**.

Notre projet consiste à réaliser, concevoir, développer, tester et valider une application de gestion réseau baser sur le protocole **SNMP**.

3.1.1 Présentation de l'organisme d'accueil

Notre projet à été réalisé au sein de la **DRGB** (direction régionale de Bejaia) qui est une branche de la SONATRACH qui sera représentée dans ce qui suit.

3.1.1.1 Vue globale de la SONATRACH

SONATRACH est la première société du continent africain, elle est classée 11eme parmi les compagnies pétrolières mondiales, 2eme exportateur de GNL et 3eme exportateur de gaz naturel, sa production globale (tous produits confondus) est de 2002 millions de tonnes.

3.1.1.2 Organigramme de la SONATRACH

la figure [3.1] illustre l'organogramme de la SONATRACH :

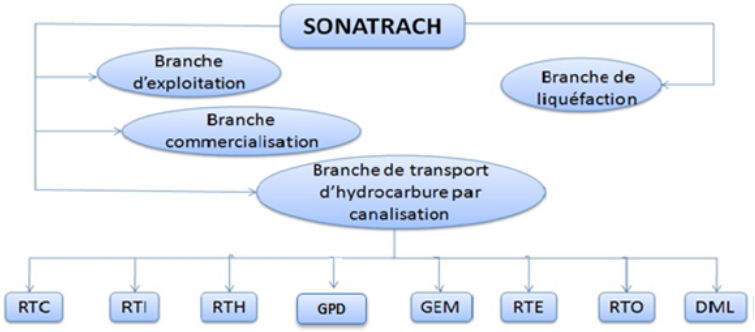

FIGURE 3.1 – organigramme de SONATRACH

- **RTC** : Région Transport centre (Bejaia).
- **RTI** : Région Transport In- Amines.
- **RTH** : Région Transport houad El-Hamra.
- **RTE** : Région Transport Est(Skikda).
- **GPF** : Gazoduc Pedro Farel (Espagne).
- **DML** : Direction Maintenance.
- **GEM** : Gazoduc Enrico Mattei.
- **RTO** : Région Transport Ouest(Arzew).

3.1.1.3 Description de la DRGB

La DRGB est située au Nord de Bejaia (arrière port), à l'entrée de la ville, sur la zone industrielle et s'étend sur une surface globale répartie :

- Terminal clôturé : 516135 m2.
- Surface couverte : 7835 m2.
- Surface occupée par les bacs : 43688 m2.
- Surface non clôturée : 2250 m2.
- Surface de stockage : 3800 m2.

La DRGB est chargée du stockage et de la livraison des hydrocarbures transportés à travers trois canalisations, qui sont de 24 pouces et de 16 pouces pour le pétrole et le condensât, de 42 pouces pour le gaz.

Sa capacité de transport est environ 14 millions de tonnes. La capacité réelle de transport est de 11 millions, dont 9 millions sont destinées a l'exportation et 2 millions de tonnes sont acheminées sur la raffinerie d'Alger.

L'effectif total de la DRGB est de 2819 employées dont 864 sont permanents et 1955 sont temporaires.

3.1.1.4 Organigramme de la DRGB

Nous montrons dans la figure [3.2] l'organigramme de la DRGB :

FIGURE 3.2 – Organigramme de la DRGB.

3.2 Analyse des besoins

Afin de garantir la réussite et l'efficacité de notre projet, il faut à ce stade du travail, définir avec précision les bordures de la solution à développer.

Ceci inclut d'une manière précise, correcte, compréhensive les différents besoins de l'utilisateur par l'élaboration de tous les diagrammes qui représentent le système.

3.2.1 Identification des besoins

Les besoins d'utilisation de notre application sont repartis en besoins fonctionnels et non fonctionnels.

3.2.1.1 Les besoins fonctionnels

Les besoins fonctionnels incluent les modules de gestion de l'application à réaliser tels que :

- **Gestion des utilisateurs** :permet d'ajouter, modifier, supprimer un autre administrateur.
- **Supervision des équipements** : permet de contrôler l'état de configuration des équipements.
- **Gestion des fautes** : permet la récupération des alarmes (traps).
- **Gestion des performances** : notre application doit permettre de gérer des rapports de chaque équipement qui décrit ses informations et aussi d'enregistrer chaque trace d'administrateur dans son historique.

3.2.1.2 Les besoins non fonctionnels

Les besoins non fonctionnels sont les exigences qui ne concernent pas spécifiquement le comportement du système, mais plutôt d'identifier les contraintes internes et externes du système tels que :

- Le code doit être clair pour permettre des futures évolutions ou améliorations.
- L'ergonomie : l'application offre une interface conviviale et facile à utiliser.
- La sécurité : l'application doit respecter la confidentialité des données.

3.2.2 Présentation du langage de modélisation

Modéliser un système avant sa réalisation permet de mieux comprendre son fonctionnement.C'est, également, un bon moyen pour maitriser sa complexité et d'assurer sa cohérence.

A travers cette section, nous allons introduire, d'une façon générale, le langage de modélisation **UML**, et nous allons présenter le processus du développement utilisé ; il s'agit de **UP** (unified process).

3.2.2.1 Définition du processus unifié (UP)

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

3.2.2.2 Définition du langage UML

Le langage UML est un langage de modélisation pseudo-formel qui propose une notation permettant de représenter graphiquement les éléments de modélisation du méta modèle.

Cette notation graphique est le support du langage UML car elle permet d'exprimer visuellement une solution orientée objet. L'aspect formel du langage UML limite les ambigüités et les incompréhensions. Bien qu'il ne soit pas un processus, le langage UML facilite une démarche d'analyse itérative et incrémentale, basée sur les niveaux d'abstraction.[4]

3.2.2.3 Les diagramme UML

UML dans sa version 02 s'articule autour de treize (13) diagrammes, chacun d'entre eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier.

Ces diagrammes sont regroupés dans deux grands ensembles : les diagrammes structurels et les diagrammes de comportement.

L'ensemble des 13 types de diagrammes UML sont illustrés dans la figure [3.3].

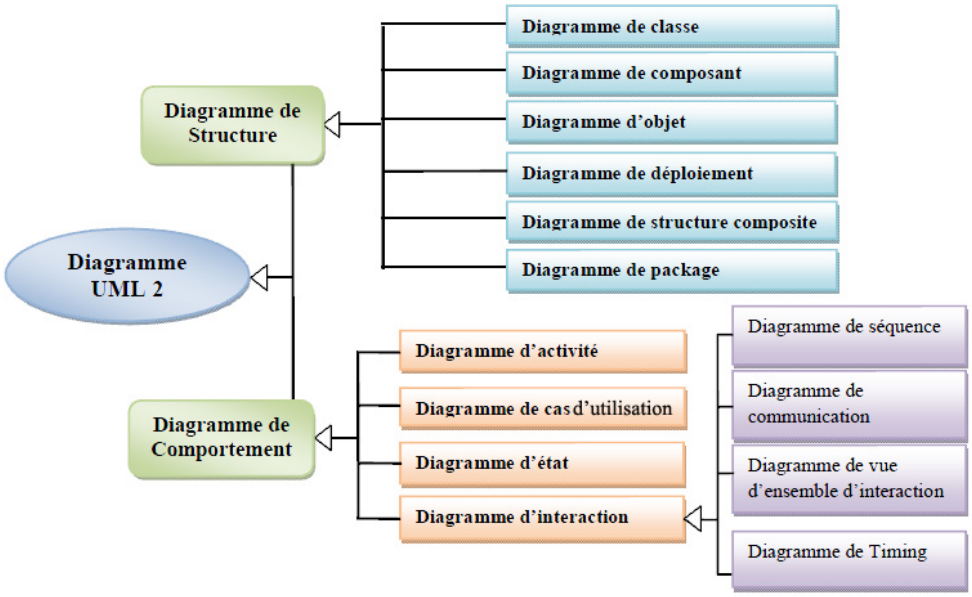

FIGURE 3.3 – les diagrammes UML.

3.2.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation (ovales) reliés par des associations (lignes) à leurs acteurs (icône du « stick man », ou représentation graphique équivalente). Chaque association signifie simplement « participe à ». Un cas d'utilisation doit être relié au moins à un acteur [10].

La figure [3.4] montre le format général d'un diagramme de cas d'utilisation :

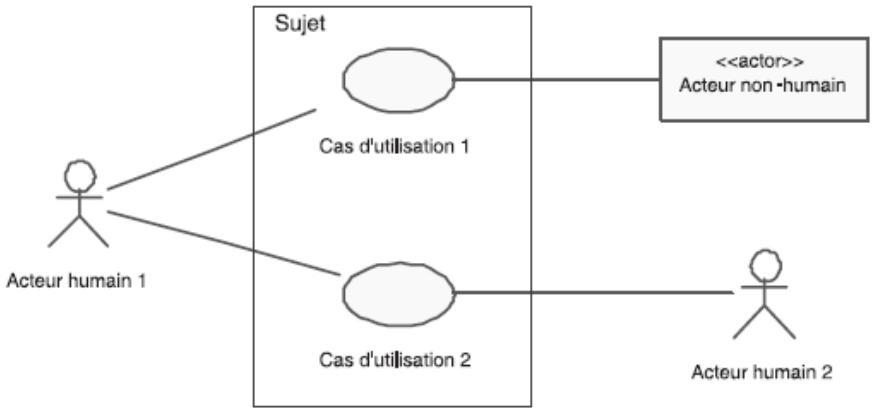

FIGURE 3.4 – formalisme de représentation du diagramme de cas d'utilisation.
[10]

3.2.3.1 Diagramme de cas d'utilisation de l'application

Le diagramme de l'ensemble des cas d'utilisation de l'application est représenté dans la figure [3.5]

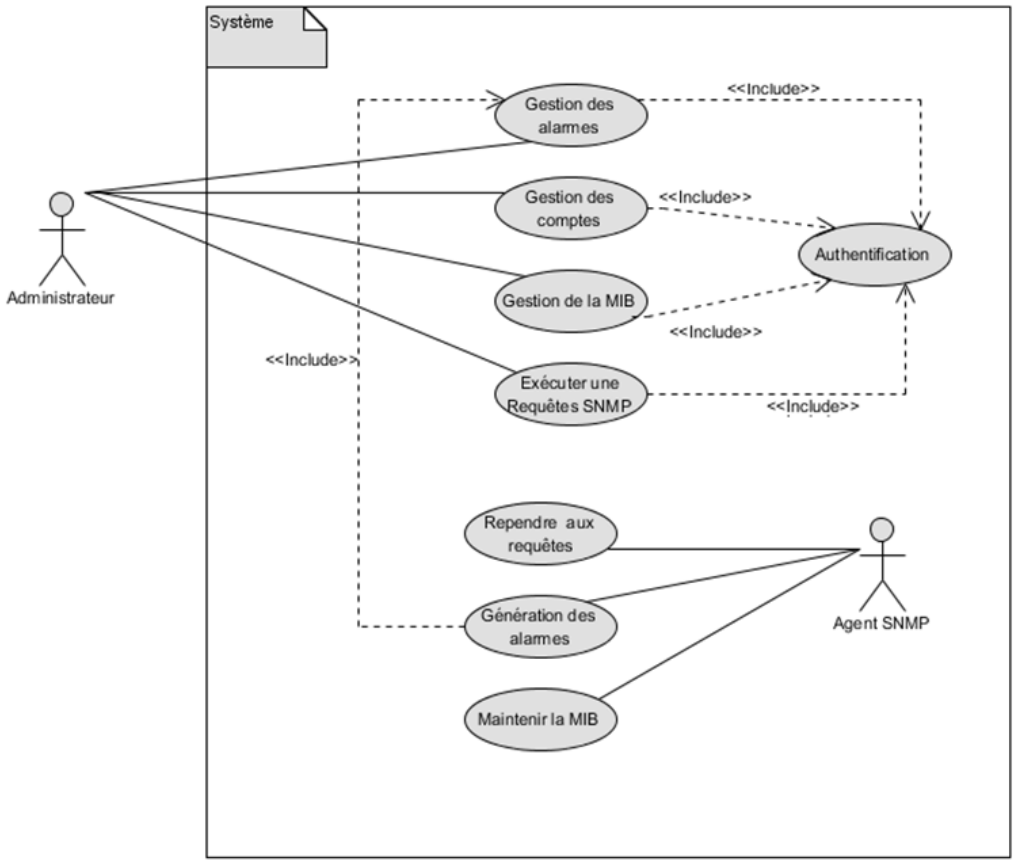

FIGURE 3.5 – Diagramme de cas d'utilisation.

3.2.3.2 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. [11]

Notre projet distingue deux acteurs qui sont les suivants :

- **L'administrateur** : qui a pour rôle de gérer l'application.
- **L'agent SNMP** : son rôle est de maintenir la MIB et d'envoyer des alarmes à l'administrateur.

3.2.3.3 Identification des cas d'utilisation

L'ensemble des cas des utilisations sont introduit dans le tableau [3.1].

Cas d'utilisation	**Acteur**	**Action**
Gestion des comptes	Administrateur	Modifier le mot de passe et le nom d'utilisateur
Gestion des traps	Administrateur	Ecoute des traps générés par l'agent.
Gestion de la MIB	Administrateur	Chargement, déchargement et consultation de la MIB.
Requêtes SNMP	Administrateur	Exécuter une requête (GET,SET,...etc.) .
Rependre aux requêtes	Agent SNMP	Rependre aux requêtes émises par l'administrateur
Génération des alarmes (trap)	Agent SNMP	Génére les alarmes en cas d'erreur.
Maintenir la MIB	Agent SNMP	Mise à jour des valeurs des objets de la MIB.

TABLE 3.1 – Identification des cas d'utilisation.

3.2.3.4 Description textuelle des cas d'utilisation

- **Cas d'utilisation** : Authentification à l'application

Desciption sommaire :

Acteurs : Administrateur.

Objectif : donner l'accès à l'application.

Description des enchainements :

Pré condition : exécuter l'application

Poste condition : ouverture de la session

Scénario nominal : l'administrateur introduit le nom d'utilisateur et le mot de passe.

Exception : Le nom d'utilisateur ou le mot de passe (voir les deux) est erroné donc l'accès au système sera refusé, le système demande alors à l'utilisateur d'introduire à nouveau son nom et son mot de passe.

- **Cas d'utilisation** : Gestion des comptes

Desciption sommaire :
Acteurs : Administrateur.
Objectif : Modifier le mot de passe et le nom d'utilisateur.
Description des enchainements :
Pré condition : exécuter l'application
Poste condition : ouverture de la session
Scénario nominal : l'administrateur choisis le bouton " changer le mot de passe " ensuite, il remplira les informations nécessaire dans le formulaire qui va apparaitre.

- **Cas d'utilisation** : Gestion des traps

Desciption sommaire :
Acteurs : Administrateur.
Objectif : Ecoute des traps générés par l'agent.
Description des enchainements :
Pré condition : Authentification à l'application
Poste condition : Ouverture de la session
Scénario nominal : l'administrateur lance l'écouteur de traps sur le port N° 162.

- **Cas d'utilisation** : Gestion de la MIB

Desciption sommaire :
Acteurs : Administrateur.
Objectif : Charger, décharger une MIB.
Description des enchainements :
Pré condition : Authentification à l'application
Poste condition : l'administrateur doit sélectionner la MIB à charger ou bien à décharger
Scénario nominal : l'administrateur demande de charger ou bien de décharger une MIB, le système lui indique l'emplacement de cette dernière, puis il la sélectionne .

- **Cas d'utilisation** : Requêtes SNMP

Desciption sommaire :
Acteurs : Administrateur.
Objectif : Consultation et mise à jour des objets de la MIB.
Description des enchainements :
Pré condition : Authentification à l'application
Poste condition : Sélectionner un objet dans la MIB ; Introduire la communauté et OID (l'identifiant de l'objet)
Scénario nominal : L'administrateur interroge l'agent sur une valeur d'un objet, en choisissant un type de requête(GET,GETNext,GETBulk, SET)

- **Cas d'utilisation** : Exécuter les requêtes

Desciption sommaire :
Acteurs : Agent.
Objectif : Exécuter les requêtes émises par l'administrateur.
Description des enchainements :
Pré condition : demande d'une valeur d'un objet par l'administrateur.
Poste condition : Réception des requêtes émises par l'administrateur.
Scénario nominal : exécuter la requête et envoyer le résultat à la station d'administration.

- **Cas d'utilisation** : Génération des alarmes

Desciption sommaire :
Acteurs : Agent.
Objectif : Envoi des notifications à l'administrateur.
Description des enchainements :
Pré condition : l'administrateur doit lancer l'écouteur de trap.
Poste condition : envoi les traps sur le port N° 162.
Scénario nominal : l'agent génère les traps ensuite il repend à l'écouteur de trap qui a été lancé par l'administrateur.

- **Cas d'utilisation** : Maintenir la MIB

Desciption sommaire :
Acteurs : Agent.
Objectif : Consultation des objets de la MIB et mise à jour des variables de la MIB.
Description des enchainements :
Pré condition : Réception d'une requête émise par l'administrateur
Poste condition : introduction de la nouvelle valeur à mettre à jour.
Scénario nominal : exécuter la requête et mettre à jour la MIB.

3.2.4 Diagramme de séquence

Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.[4] le format général de diagramme de séquence est illustré dans la figure [3.6]

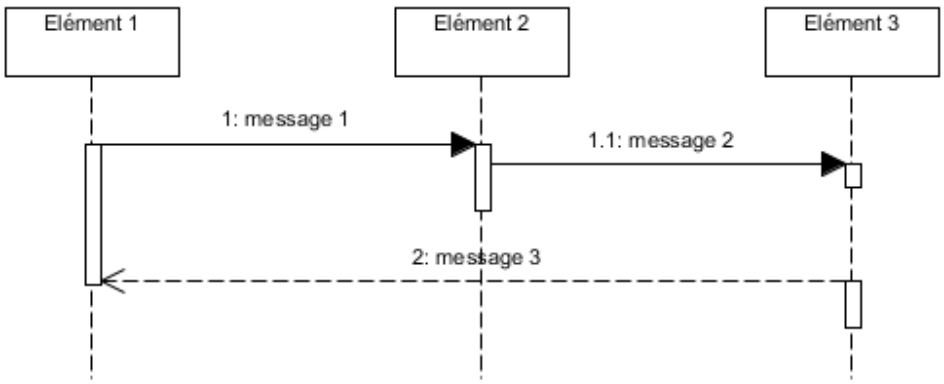

FIGURE 3.6 – formalisme de représentation du diagramme de séquence.

- **Diagramme de séquence du cas d'utilisation «Authentification »**

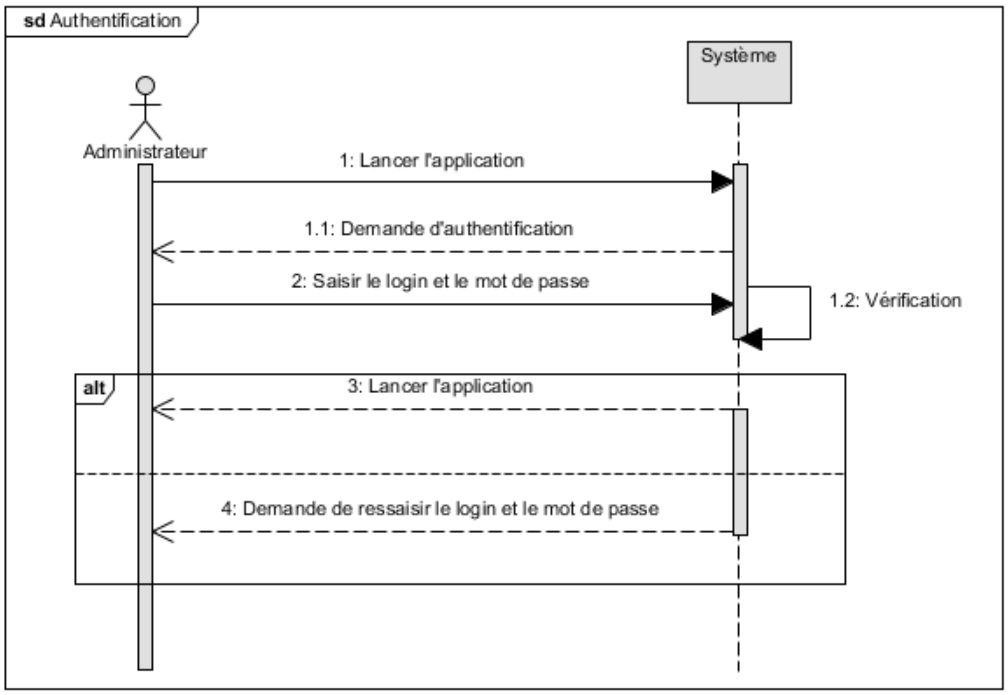

FIGURE 3.7 – Diagramme de séquence du cas d'utilisation «Authentification »

- **Diagramme de séquence du cas d'utilisation «Gestion des traps»**

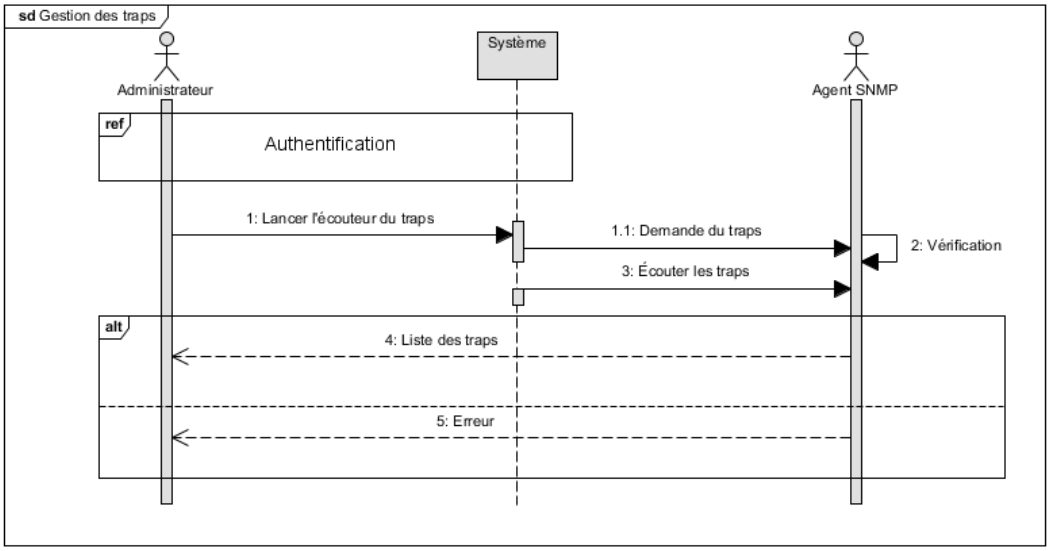

FIGURE 3.8 – diagramme de séquence du cas d'utilisation « Gestion des traps »

- **Diagramme de séquence du cas d'utilisation «Chargement de la MIB»**

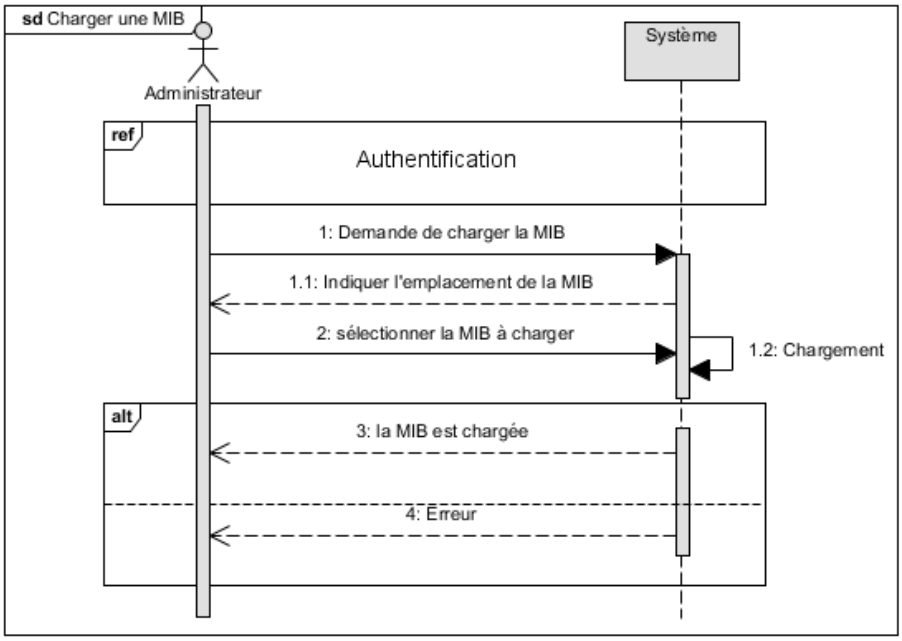

FIGURE 3.9 – diagramme de séquence de cas d'utilisation « Chargement de la MIB »

- **Diagramme de séquence du cas d'utilisation «Déhargement de la MIB»**

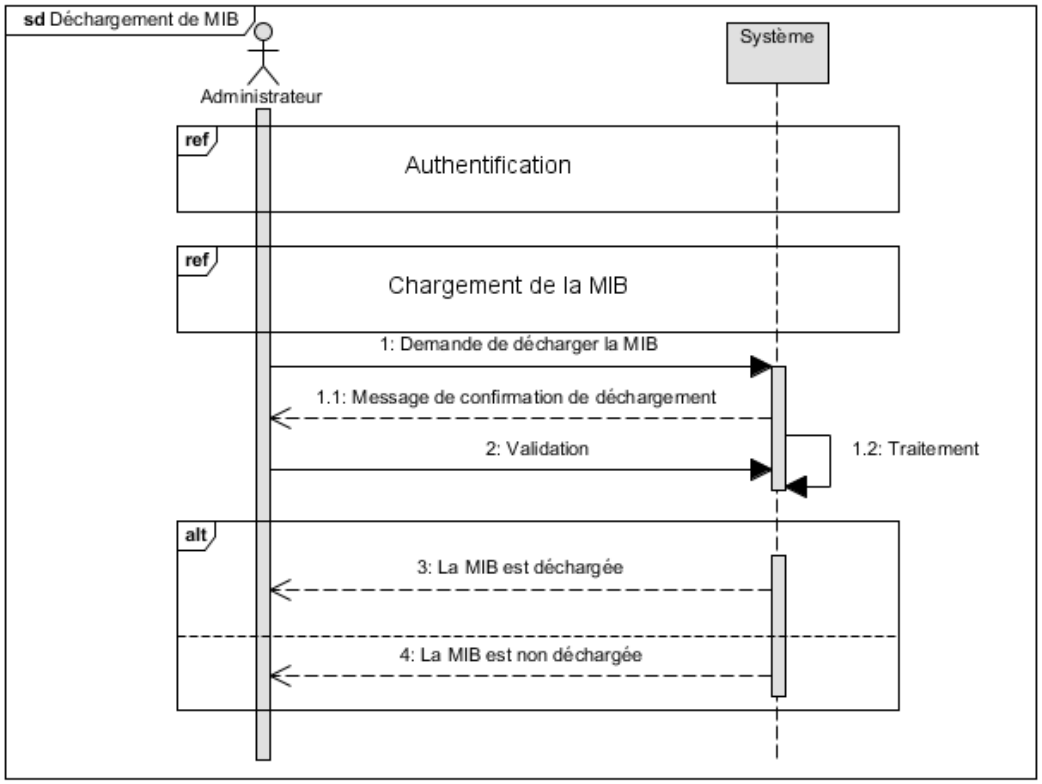

FIGURE 3.10 – diagramme de séquence de cas d'utilisation « Déchargement de la MIB »

- **Diagramme de séquence du cas d'utilisation «Exécuter une requête SNMP»**

les requêtes SNMP sont de deux types : les requêtes sécuriseés et non sécurisées.

  - **Diagramme de séquence de requêtes SNMP non sécurisées**

- **Requête GET**

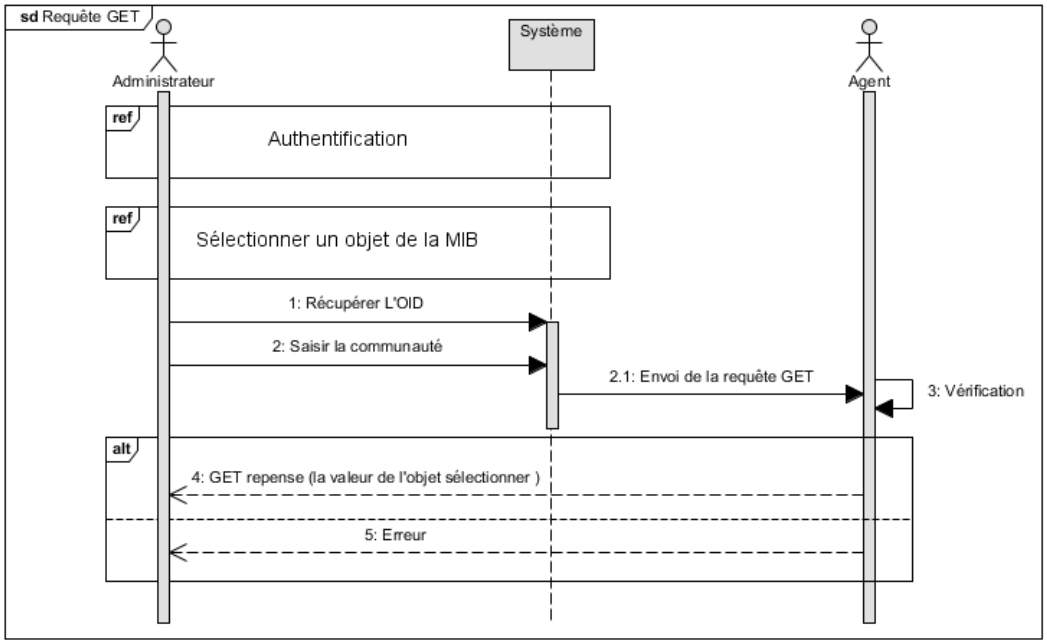

FIGURE 3.11 – diagramme de séquence de cas d'utilisation «Requête GET non sécurisée »

- **Requête GETNext**

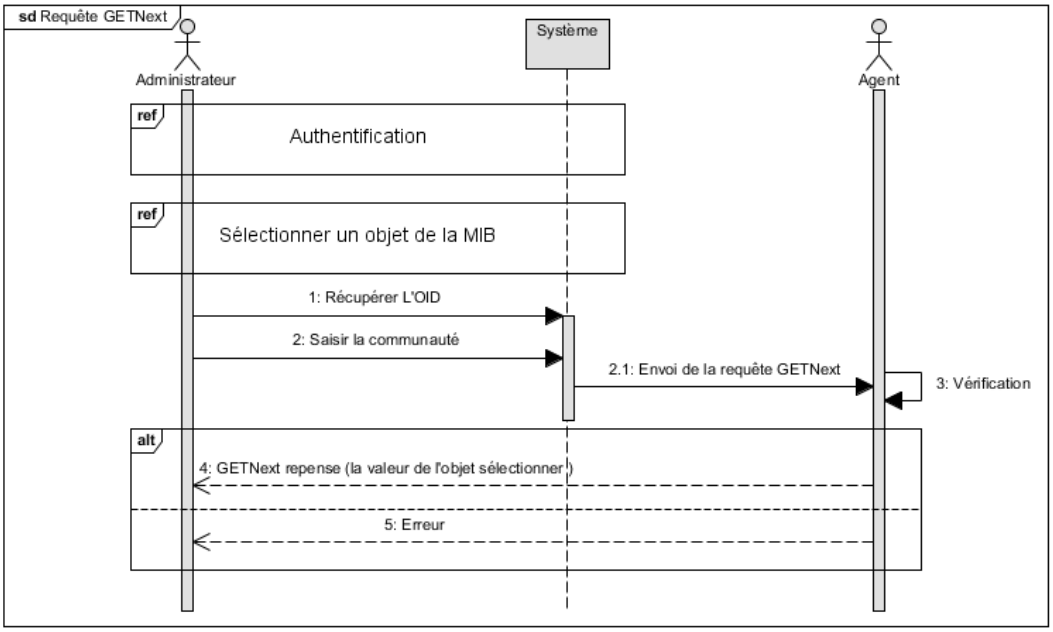

FIGURE 3.12 – diagramme de séquence de cas d'utilisation «Requête GETNext non sécurisée».

- **Requête GETBulk**

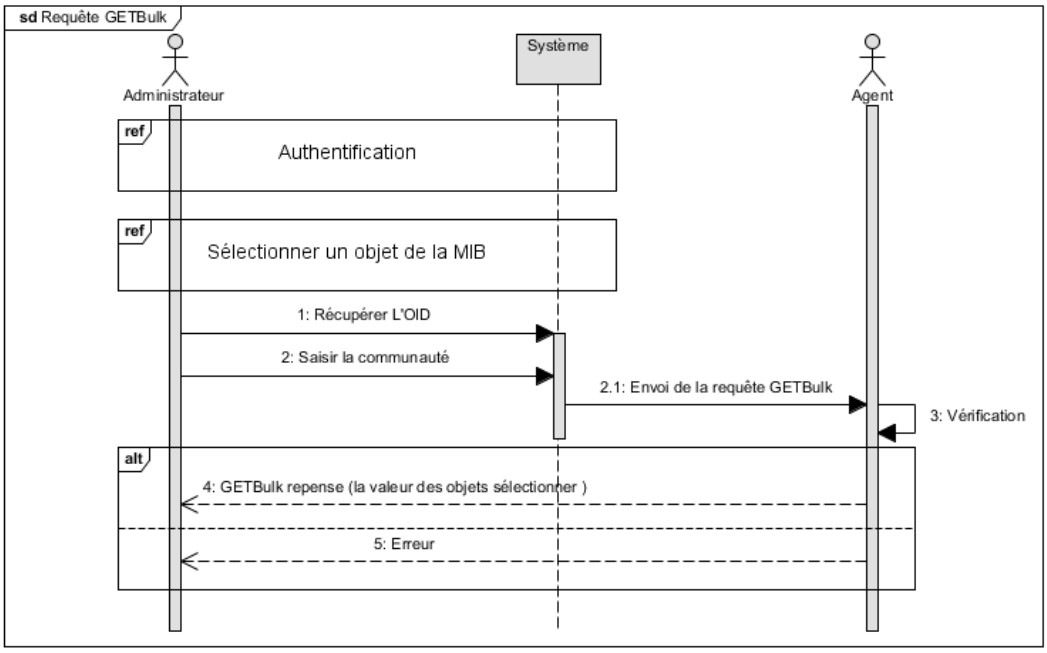

FIGURE 3.13 – diagramme de séquence de cas d'utilisation «Requête GETBulk non sécurisée».

- **Requête SET**

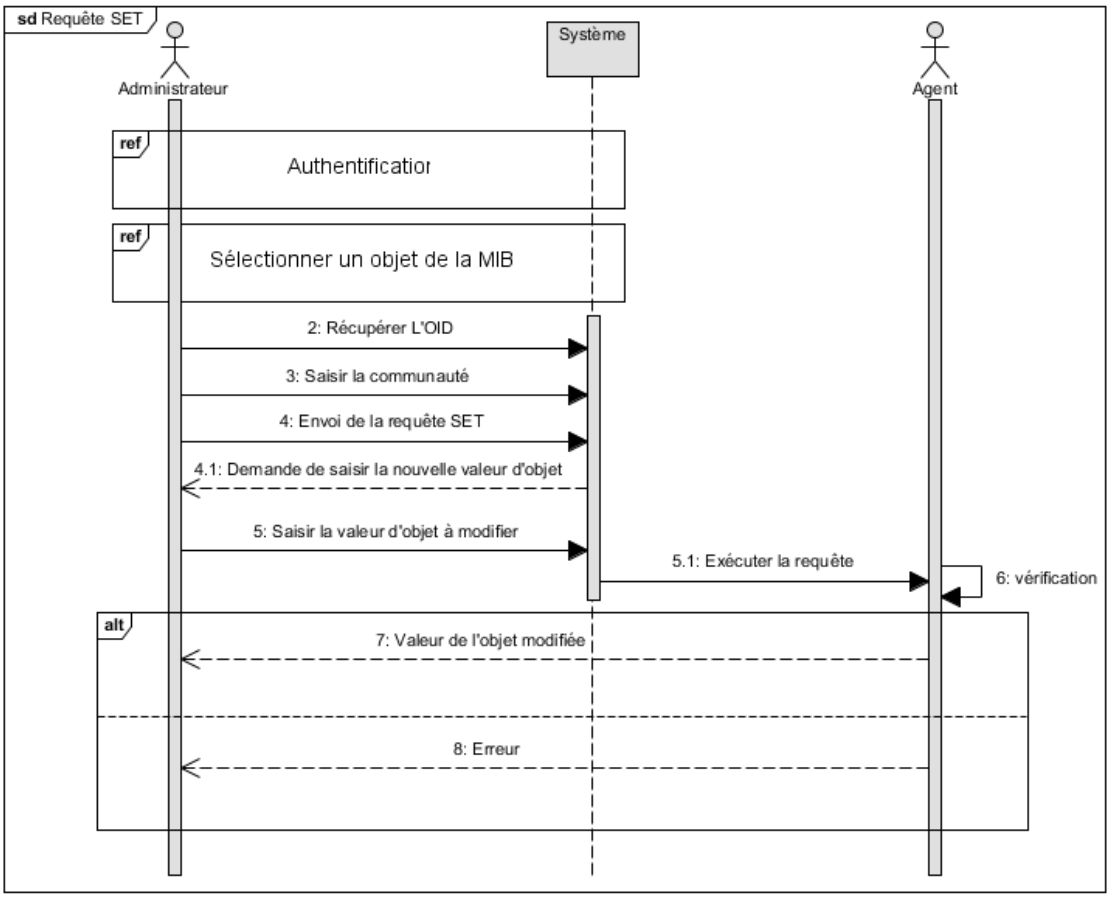

FIGURE 3.14 – diagramme de séquence de cas d'utilisation «Requête SET non sécurisée».

- **Diagramme de séquence de requêtes SNMP sécurisées**

- **Requête GET**

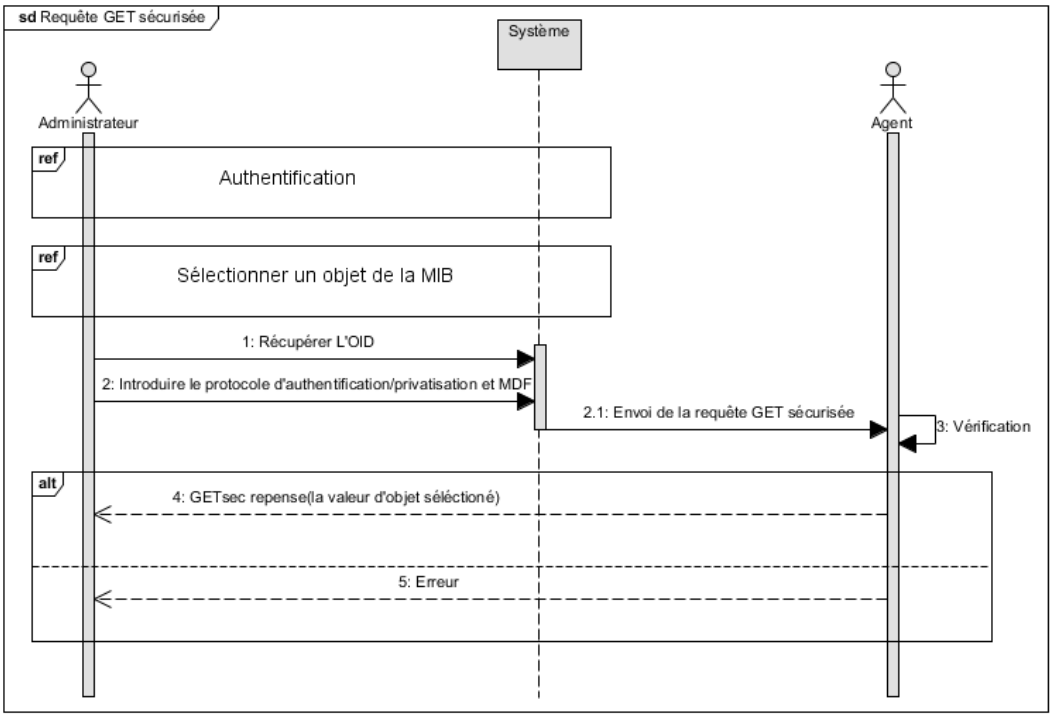

FIGURE 3.15 – diagramme de séquence de cas d'utilisation «Requête GET sécurisée »

- **Requête GETNext**

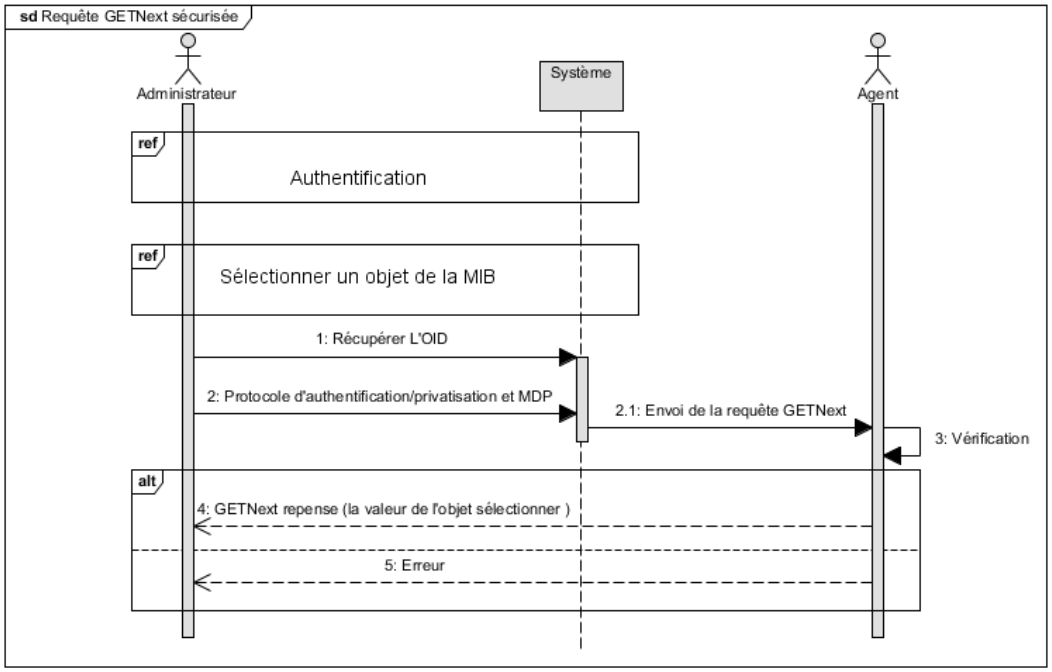

FIGURE 3.16 – diagramme de séquence de cas d'utilisation «Requête GETNext sécurisée».

- **Requête GETBulk**

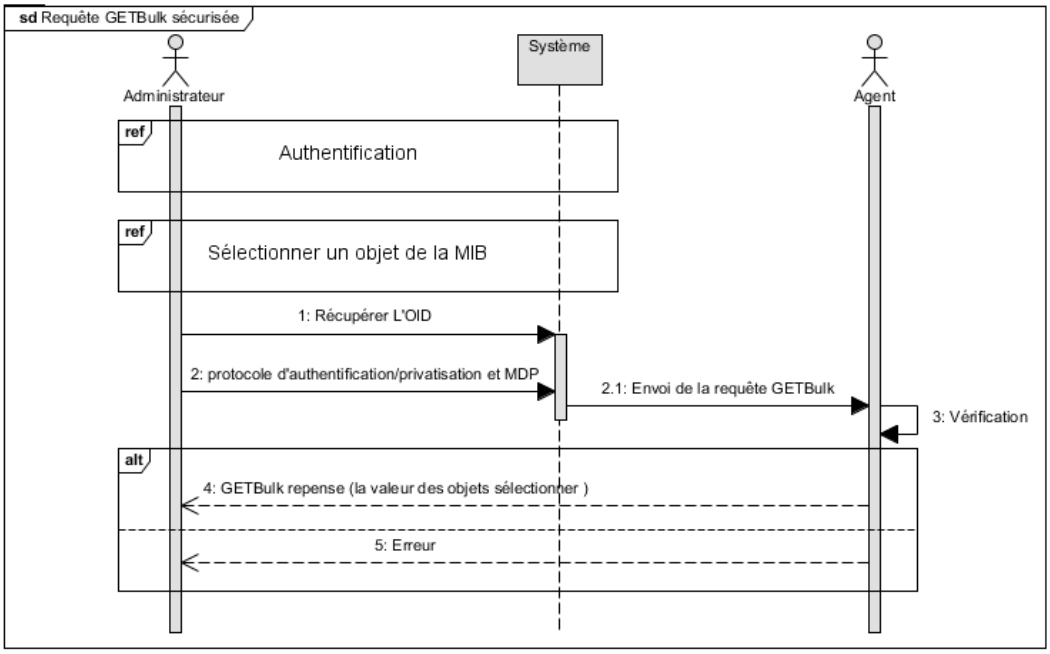

FIGURE 3.17 – diagramme de séquence de cas d'utilisation «Requête GETBulk sécurisée».

- **Requête SET**

3.2.5 Diagramme d'activité

Ce diagramme donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les fluts de contrôle et les fluts de données.[4]

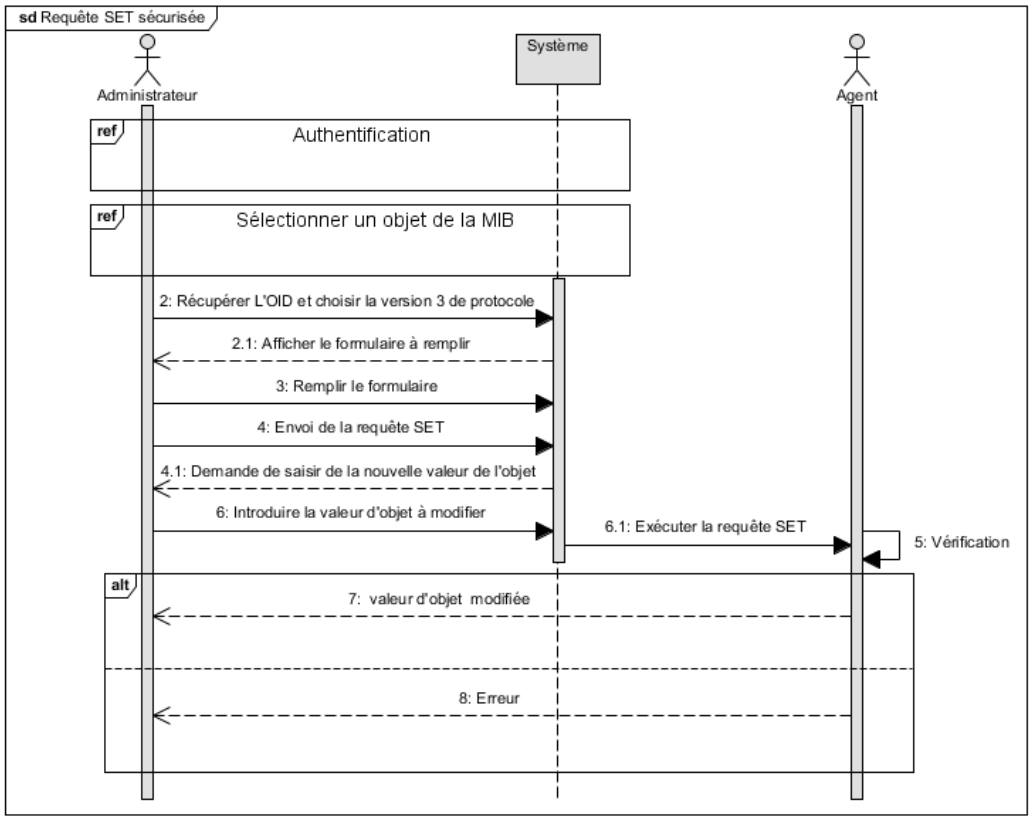

FIGURE 3.18 – diagramme de séquence de cas d'utilisation «Requête SET sécurisée».

le formalisme de ce diagramme est illustré dans la figure [3.19] :

Pour que l'utilisateur accède à l'application, il doit d'abord s'authentifier, après il peut effectuer l'ensemble des activités de notre système qui sont les suivantes :

- Charger une MIB.
- Sélectionner un objet de la MIB.
- Sélectionner une requête à exécuter.
- Saisir la communauté.
- Envoi de la requête choisie.
- Afficher le résultat.

L'ensemble des activités qui sont effectuées par l'utilisateur sont présentées dans le diagramme d'activité qui est illustré dans la figure [3.20] :

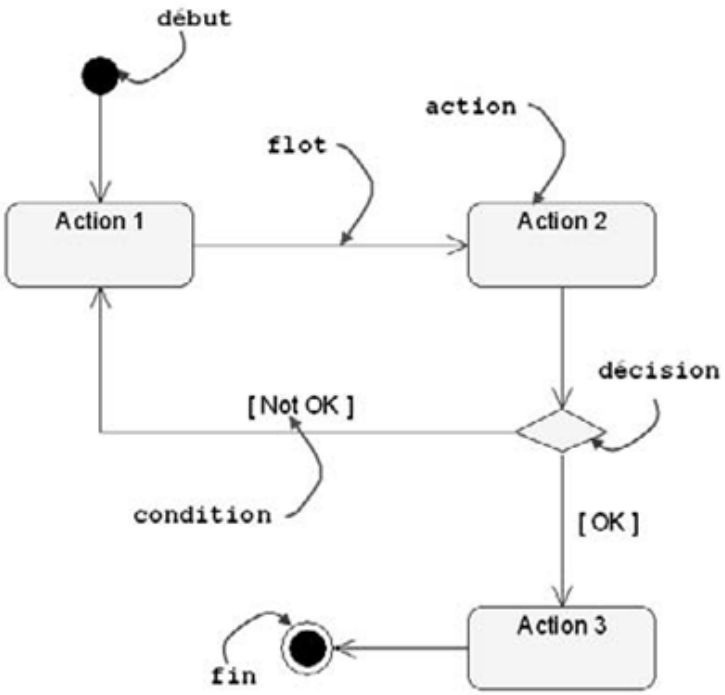

FIGURE 3.19 – formalisme de représentation du diagramme d'activité. [10]

3.3 Conception

La conception est la dernière phase de la modélisation avec UML. Après la modélisation des besoins et l'organisation de la structure de la solution.

La conception vient construire et documenter précisément les classes, les interfaces, les tables et les méthodes qui constituent le codage de la solution.

Dans cette partie nous allons concevoir les classes, les associations, les attribues et les opérations.

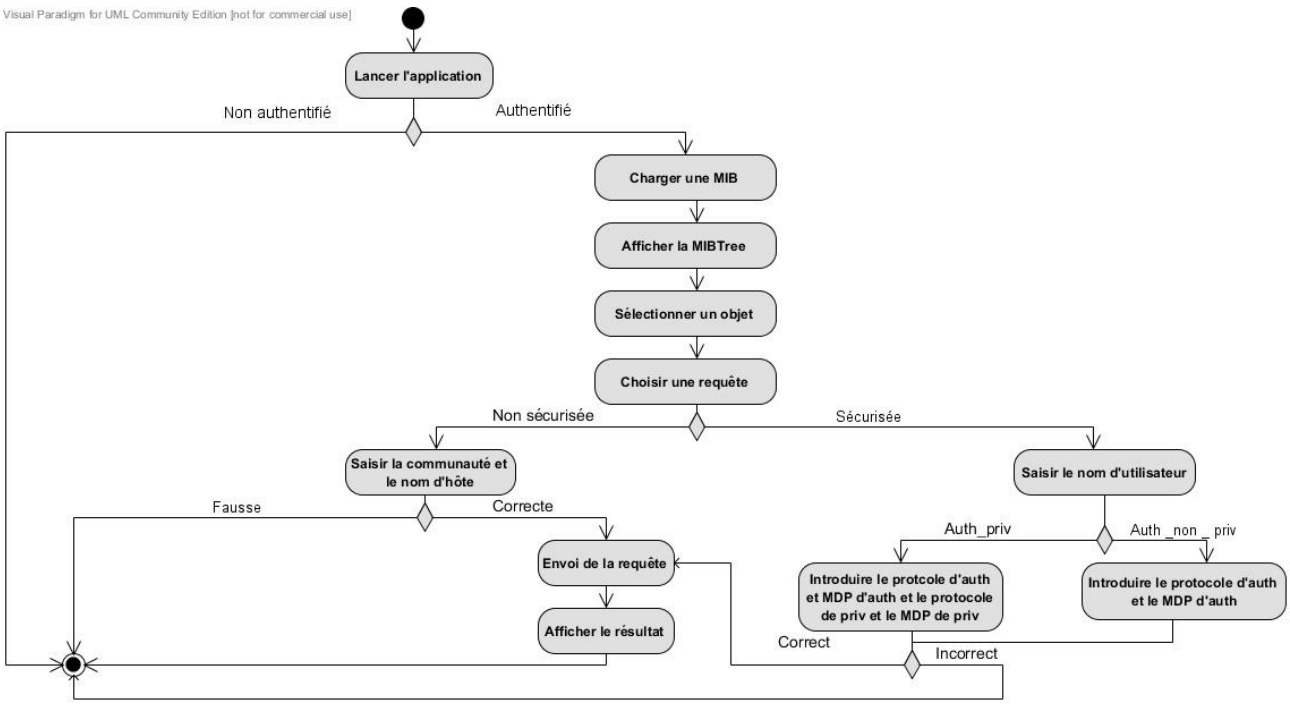

FIGURE 3.20 – Diagramme d'activité de l'application.

3.3.1 Dictionnaire de données

Le tableau [3.2] représente les classes et ses attributs, ainsi que leurs types et les opérations effectuées par chaque classe :

classe	attributs	type	Opération
Manager	Nom	String	Authentifier ()
	Mot de passe	String	
	Prénom	String	
UI-fenêtre	Version	String	Getcomp()
	Hote	String	getOid()
	Port	String	getHot()
	Communauté	String	setRéponse()
	OID	String	setCharge()
			exec_get()
			exec_getnext()
			exec_getbulk()
			exec_set ()
Get			Execute()
Trap			Received_trap()
			Lancecoute()
Getnext			Execute()
Getbulk			Execute()
Set			Execute()
MibArbre			SetNod()
			Setdesc()
			ajoutMib()
			Decharg()
ChargMib			Chargmib()
			Getmib()
			Setmib()

Demo_graph	Oid url hot interval	String String String Integer	Setvalue ()
Line_graph	Oid Url hot interval	String String String Integer	Setvalue()
DialogSet	Nouv-val communauté	String String	
Dialogue bulk	Nbre	String	
Authentification	Login Motpasse	String String	

TABLE 3.2 – Dictionnaire de données.

3.3.2 Diagramme de classe

Le diagramme de classes est considéré comme le plus important de tous les autres diagrammes lors de la modélisation orientée objet en langage UML.

Il est indispensable lors d'une telle modélisation, il montre la structure interne du système à modéliser, il contribue ainsi à clarifier l'axe statique de la modélisation basée sur UML. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation.

La figure [3.21] illustre le diagramme de classe de notre application :

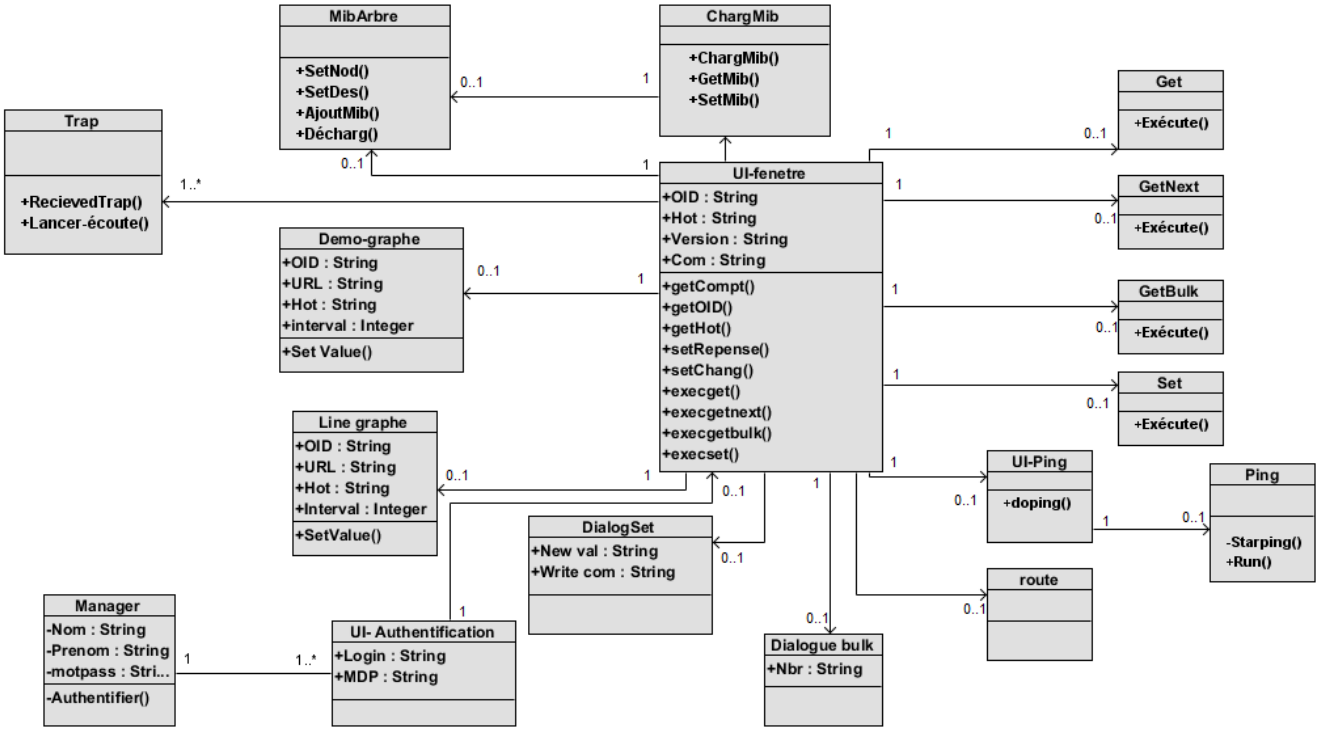

FIGURE 3.21 – Diagramme de classe de l'application.

Conclusion

La conception d'un système quelconque est la tâche qui consiste à déterminer l'architecture de l'application avant de la mettre en œuvre.

Dans ce chapitre nous avons déterminé et spécifié l'ensemble des besoins fonctionnels et non fonctionnels de l'application qui nous a permis de réaliser le diagramme de cas d'utilisation.

Ensuite, nous avons détaillé les scénarios de ces cas d'utilisation et l'ensemble des traitements impliqués pour la réalisation des diagrammes de séquence.

Enfin, pour compléter cette partie, nous avons conçu les classes et ses attributs pour accomplir le diagramme de classe.

4

Implémentation et Tests

Introduction

Après avoire étudier les différents besoins de l'utilisateur et modéliser le système, nous pouvons alors entreprendre la dernière activité du Processus Unifié qu'est de même composé de deux parties (implémentation et test), ayant comme objectif d'aboutir à un produit final, exploitable par les utilisateurs.

Dans cette phase, nous allons présenter l'environnement de développement que nous avons utilisé, l'architecture matérielle mise en place, implémenter tous les cas d'utilisation, et enfin les tester.

4.1 Implémentation

Cette section présente les outils disponibles pour l'implémentation des applications de gestion réseau, de même que l'implémentation physique de notre application en spécifiant la technologie utilisée.

4.1.1 API (*Application Programming Interface*)

Plusieurs logiciels de gestion SNMP ont été mis en œuvre, tel que Net-SNMP qui est un ensemble d'outils permettants d'utiliser et de déployer le protocole SNMP (v1,v2,v3), ainsi que AdventNet qui représente un ensemble complet d'outils pour créer un environnement simulé.

L'API que nous avons employée pour la réalisation de notre application est SNMP API AdventNet.

L'API AdventNet SNMP est une API commercialisée, cependant, elle peut être utilisée pour usage académique,ou employée pour développer des application de gestion de réseau.

Les développeurs d'applications de gestion de réseau peuvent utiliser la bibliothèque SNMP d'AdventNet pour construire des applet, des composants, et des applications réparties d'EJB, de CORBA, et de RMI.

la bibliothèque fournit les fonctions et les composants les plus généralement utilisées pour rendre le développement plus simple.

4.1.2 Environnement de développement de l'application

Nous avons développé cette application dans la plate forme Windows 7, a l'aide de l'Environnement de développement intégré **NETBEANS**. Ce qui rend le déploiement de l'application facile dans la plate forme Windows.

Notre choix s'est porté sur le langage **JAVA**, l'intérêt porté à ce langage est motivé par ses caractéristiques et sa portabilité.

4.1.2.1 Présentation de NETBEANS

NetBeans est un environnement de développement intégré, gratuit à l'usage, se concentrant principalement sur la simplification de développement d'applications Java.

Il fournit un support pour tous les types d'applications Java, depuis le client riche jusqu'aux applications d'entreprises multicouches, en passant par les applications pour les mo-

biles supportant Java.

4.1.2.2 Présentation du langage java

Java a été conçue par James Gosling en 1994 chez Sun. L'idée était d'avoir un langage de développement simple, portable, orienté objet, interprété.

Java reprend la syntaxe de C++ en le simplifiant. Java offre aussi un ensemble de classes pour développer des applications de types très variés (réseau, interface graphique, multitâches, etc.) Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

- **Java est interprété** :le code source est compilé en pseudo code ou byte code puis exécuté par un interpréteur Java.
- **Java est orienté objet** : comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application.
- **Java assure la gestion de la mémoire** : l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.
- **Java est économe** : le pseudo code à une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.
- **Java est multitâche** : il permet l'utilisation de threads qui sont des unités d'exécution isolées. La JVM, elle même, utilise plusieurs threads.

4.2 Tests et interfaces de l'application

Dans cette partie nous allons présenter l'application ainsi que son architecture et ses fonctionalités d'une part, d'autre part, nous allons tester les cas d'utilisation et illustrer quelques interfaces de l'application.

4.2.1 Présentation de l'application

Notre application permet de superviser, contrôler l'état de configuration, et récupéré les traps généré par l'agent SNMP.

Les princpales fonctionnaliés de l'application sont :

- Chargement et déchargement d'une MIB.
- Choisir la version de SNMP à utiliser.
- Ecoute des traps émis par l'agent SNMP.
- Afficher la table d'un objet.
- Afficher la description des objets de façon détaillée.
- Effectuer des opérations de consultation : GET, GetNext, GetBulk.
- Effectuer des mises à jour : SET.

4.2.1.1 Activation de l'agent SNMP

Pour utiliser cette application il faut d'abord configurer l'agent SNMP en suivant les étapes :

- Ouvrir l'Assistant Composants de Windows : Cliquez sur Démarrer, pointez sur Paramètres ;
- Cliquer sur le Panneau de configuration, double-cliquez sur programmes et fonctionnalités ;
- Puis cliquer sur activer ou désactiver des fonctionnalités Windows ;
- Dans Composants, cliquez sur Outils de gestion et d'analyse (sans activer ni désactiver la case à cocher correspondante), puis cliquez sur Détails ;
- Activer la case à cocher SNMP (Protocole simplifié de gestion de réseau), puis cliquer sur OK ;
- Cliquer sur Suivant. (SNMP démarre automatiquement à la fin de l'installation).

4.2.1.2 Architecture de l'application

L'architecture de l'application vise à prendre en compte toutes les fonctionnalités nécessaires à la construction de l'application. Ses fonctionnalités sont divisées en trois parties :

- **La partie présentation** : Elle représente l'interface utilisateur de l'application ; ses principales fonctions sont :
  - Effectuer des requêtes de consultation et de mise à jour des objets.
  - Ecoute les tarps.
- **La partie gestion de réseau** : Elle s'occupe de toutes les opérations de gestion réseau par SNMP dont elle permet de :
  - Lancer des requêtes SNMP.

– Lancer ou arrêter le processus d'écoute de traps à l'aide de gestionnaire de trap.

- **La partie gestion de la MIB** :C'est la partie qui a pour rôle la gestion de la MIB, elle prend en charge toutes les opérations qui peuvent être effectuées sur cette dernière.

L'architecture de l'application est illustrée dans la figure [4.1] :

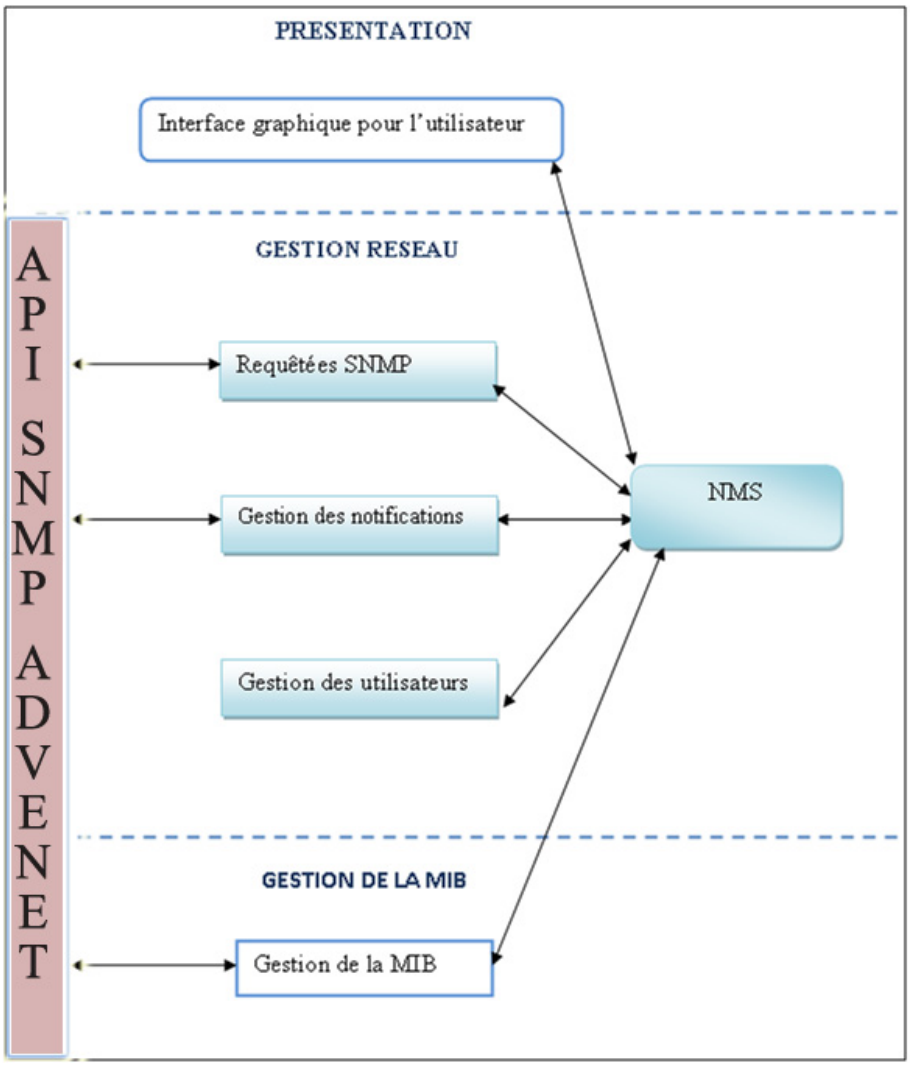

FIGURE 4.1 – Architecture de l'application.

4.2.1.3 Interfaces de l'application

Dans cette section nous allons présenter quelques interfaces de l'application.

- **l'interface principale de l'application**

  Cette interface apparait après l'authentification , c'est l'interface principale de l'application, dont l'administrateur effectue le reste des fonctionnalités.

  La figure [4.2] illustre l'interface principale de l'application.

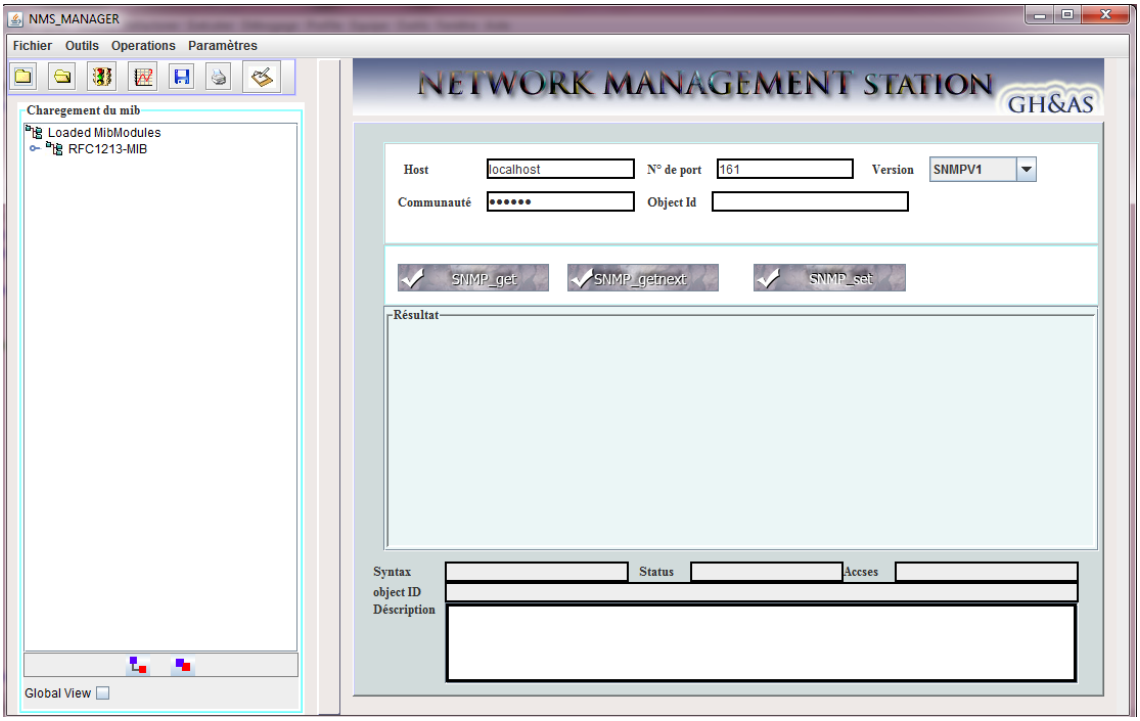

FIGURE 4.2 – Interface principale de l'application.

- **Teste de la requête GET sur la variable SysName.**

  Pour que l'administrateur effectue la requête GET, il doit d'abord parcourir la table du mib pour sélectionné une variable dans le but de récupéré sa valeur auprès de l'agent SNMP.

La figure[4.3] illustre la sélection d'une variable de la table MIB :

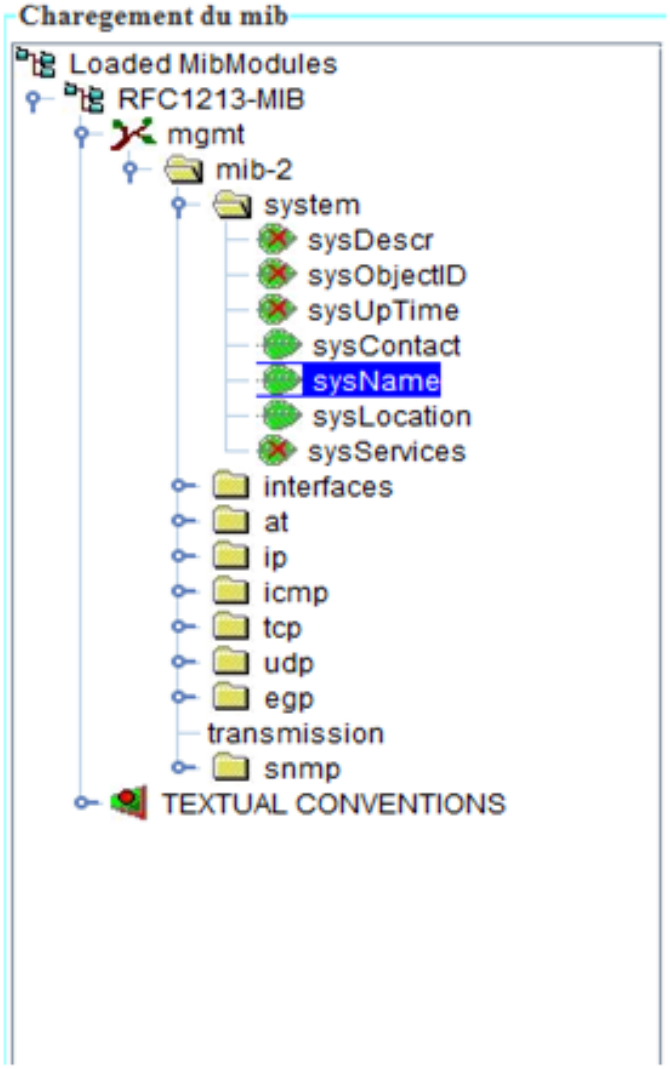

FIGURE 4.3 – Sélection d'une variable de la table MIB.

L'OID associer à cette variable est : ***.iso.org.dod.ineternet.mgmt.mib2.system.SysName***.

Le résultat d'exécution de la requête GET est exprimé dans la figure[4.4]

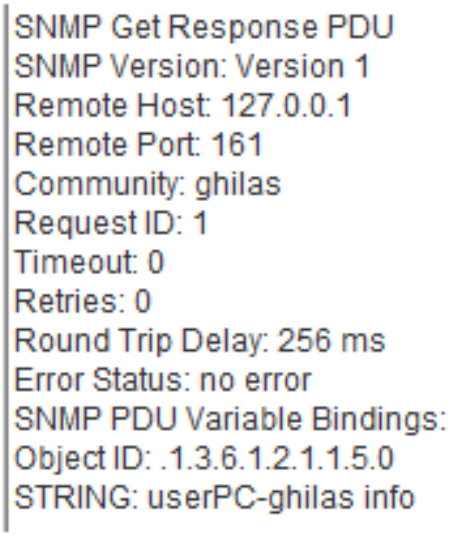

FIGURE 4.4 – Le résultat d'exécution de la requête GET.

- **Test de la requête SET sur la valeur SysName**

  L'exécution de la requête SET, nous exige de suivre la même procédure que l'exemple précédent pour sélectionner une variable du MIB.

  Ensuite nous introduisons la nouvelle valeur de la variable sélectionnée , et le nom de la communauté d'écriture.

  La figure [4.5] illustre l'interface qui permet d'introduire la nouvelle valeur de la variable : ***.iso.org.dod.ineternet.mgmt.mib2.system.SysName***

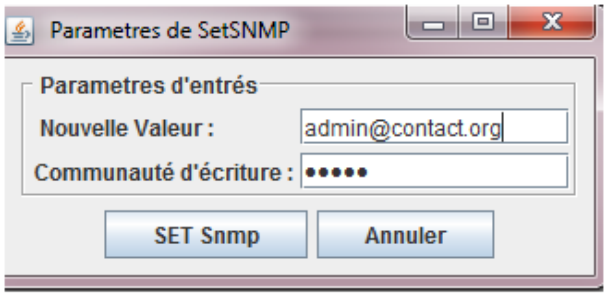

FIGURE 4.5 – fenêtre de mise à jours .

La confirmation du mis à jour est exprimer dans la figure[4.6] :

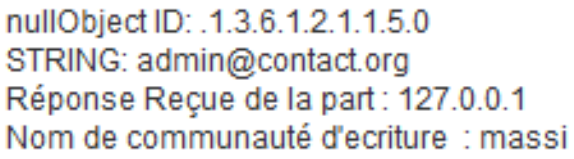

FIGURE 4.6 – Résultat de mise à jour de la requête SET.

Conclusion

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement de développement, à ce stade nous avons achevé l'implémentation et les tests de tous les cas d'utilisation, tout en respectant la conception élaborée. En d'autres termes, nous détenons la version finale du logiciel, installée dans notre environnement de développement.

Conclusion et perspectives

il n'y a pas des problèmes qu'on se pose,
il y a des problèmes qui se posent, il n'y a pas de problèmes résolus,
il y a des problèmes plus ou mois résolus
*** Henri Poincaré***

Les réseaux sont devenus indispensables au bon fonctionnement de nombreuses entreprises et administrations. Tout problème ou panne peut avoir de lourdes conséquences aussi bien financières qu'organisationnelles. La supervision des réseaux est alors nécessaire et indispensable. Elle se base principalement sur le protocole **SNMP** qui est l'objet de ce mémoire.

Nous somme parvenus par le biais de ce projet, à implémenter une application de gestion des réseaux exploitant le protocole SNMP, ce dernier a été développé pour faciliter l' administration des réseaux et qu'a l'aide des requêtes SNMP simples (**Get**, **Set**,...etc.) et la remontée d'informations par **Trap SNMP**, nous pouvons maintenir l'état du réseau.

Nous avons débuté notre travail par des généralités sur les concepts de base de la gestion des réseaux.

Dans le but de mieux cerner la tâche, nous avons fait une étude du protocole SNMP dont nous avons spécifié son architecture et son fonctionnement, ainsi que nous avons donner un aperçu sur ses versions et la sécurité donnée par **SNMPv3**.

La phase d'analyse et conception a été le cœur du développement ; au cours de cette dernière, nous avons essayé de structurer et définir les besoins attendus du futur système, il s'agissait de formuler, d'affiner et d'analyser la plus part des cas d'utilisation via des diagrammes UML.

Enfin, nous étions arrivés à la dernière phase du Processus où il s'agissait d'implémenter et tester l'application par rapport aux cas d'utilisation conçus. La version exécutable du

système est l'élément principal à livrer à l'issu de cette étape.

Nous avons pu réaliser l'essentiel du système, mais les perspéctives restent nombreuses et nous pouvons citer quelques une :

- Faciliter l'accès à notre application en utilisant une interface HTTP et un navigateur Web.Cela permettra à l'administrateur d'utiliser notre application à partir de n'importe quelle station du réseau en utilisant un navigateur Web.
- Améliorer l'application en ajoutant la sécurité (utilisation de la version 3 du protocole).
- Restreindre l'utilisation des fonctions de gestion aux administrateurs autorisés pour des raisons de sécurité.

Bibliographie

[1] A.LEINWARD ;K.FANG. *Network Management, a practical perspective.* 1993.

[2] G.PUJOLLE. *les réseaux.* septembre, 2005.

[3] H.HAITHMEN. *Utilisation du protocole SNMP pour la gestion à distance d'une interface radio par paquets.* Canada, Février, 1998.

[4] J.GABAY. *UML 2 analyse et conception.* Dunod, Paris 2008.

[5] J.Osmalaskyj. *Gestion et sécurité des réseaux informatiques.* synthèse deuxième edition, liege 2009-2010.

[6] kevin J.Schmidt, D. R. *Essetiel SNMP.* septembre, 2005.

[7] M.ARZEKI. *conception et réalisation d'une application de gestion de réseau à base de composants répartis.* Canada,juillet, 2000.

[8] M.KAHANI ;P.BEADLE. *Decentralized Approaches for Network Management.* 1997.

[9] O.CHARKAOUI. *télécommunication.* chanliere/Mc-Graw Hill, 1998.

[10] P.ROQUES. *UML 2 par la pratique étude de cas et exercices corrigés.* septembre 2006.

[11] P.ROQUES ;F.VALLE. *UML 2 en action de l'analyse des besoins à la conception.* 2007.

[12] T.BRITCH ;M.VOLAND. *Les outils d'administration et de supervision réseau L'exemple de Nagios.* Décembre, 2004.

[13] Tuang, N. M. *les protocoles pour la gestion des réseaux informatiques.* Hunoi, juillet 2005.

[14] W.STALINGS. *SNMP,SNMPV2 and CMIP -The pratical guide to Network management standards.* Addison Wesly, avril 1993.

[15] X.701, I.-T. *Information Technology - Open Systems Interconnection Systems Management Overview.* 1992.

Annexes

1. **IETF**

   L'internet Engineering Task Force, abrégée IETF, est un groupe informel, international,ouvert à tout individu qui participe à l'élaboration de standard Internet. L'IETF produit la plupart des nouveaux standards d'Ineternet.

2. **RFC**

   Les **RFC** (*Request For Comments*) sont un ensemble de documents qui font référence auprès de la Communauté Internet et qui décrivent, spécifient, aident à l'implémentation, standardisent et débattent de la majorité des normes, standards, technologies et protocoles liés à Internet et aux réseaux en général.

   Nous allons définir quelques RFC utilisées :

   – **RFC 1213 :** Management Information Base II.
   – **RFC 1155 :** Structure of Management Information.
   – **RFC 2571 :** Architecture for SNMP Frameworks.
   – **RFC 1212** : Concise MIB Definitions.
   – **RFC 1907** : MIB for SNMPv2.
   – **RFC 2115** : Frame Relay DTE Interface Type MIB.
   – **RFC 1215** : A Convention for Defining Traps (Informations).
   – **RFC 2578** : Structure of Management Information.

3. **MIB-2**

La RFC 1213 décrit le module Mib-2 qui recense un nombre d'objets de base devant être supportés par tout agent SNMP, tel que : Nom de l'administrateur, localisation, liste des interfaces réseau, vitesse, octets transmis...etc.

Groupe	OID	Description
Système	(mib-2 1)	Système géré : son nom, le type d'équipement,etc.
Interface	(mib-2 2)	Interfaces réseau.La table ifTable contient la description,l'état et les statistiques des interfaces.
at	(mib-2 3)	Address Translation : atTable contient la table des adresses MAC (Media Access Control) et réseau.Ce groupe est peut utilisé(deprecated) car il est remplacé par les équivalents spécifiques à un protocole réseau.
IP	(mib- 2 4)	IP : configuration générale, statistiques, table d'adresse (ipTable), de routage (ipRouteTable), table ARP (Address Resolution Protocol) (ipNet-ToMediaTable). ipRouteTable (21) est en pricipe remplacée par ipForward (24) "défini dans la RFC 1354" car indexée par l'adresse de destination et donc ne supportant pas les routes multiples
ICMP	(mib- 2 5)	ICMP (Internet Control Message Protocol) : statistiques.
TCP	(mib- 2 6)	TCP : configuration, statistiques générales et table des connexions (tcp-ConnTable).
UDP	(mib- 2 7)	UDP : statistiques et table des *listeners* (udpTable).
EGP	(mib- 2 8)	EGP (Exterior Gatway Protocol) : statistiques et table des voisins (egpNeighTable)
CMOT	(mib- 2 9)	CMIP over TCP/IP : obsolète. Seul l'OID est reservé dans mib-2.
Transmission	(mib- 2 10)	Ce groupe prévu pour raccrocher d'autres modules de MIB qui concernent des médias de transmissions plus spécifiques qui viennent compléter les informations contenues dans le groupe interface. Par exemple : dot3(7) décrit le media Ethernet, tandis que dot5(9) décrit Token-ring.
SNMP	(mib- 2 11)	SNMP : statistiques.