Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Virtuelle de Tunis

UVT
جامعة تونس الافتراضية
UNIVERSITÉ VIRTUELLE DE TUNIS

MEMOIRE
DE STAGE DE FIN D'ETUDES

Pour l'obtention du

«Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)»

Présenté par :
Hatem KAHLAOUI

Etude et Développement d'une Application de supervision du réseau de l'UBCI

Encadreurs : Mme Hela Boucetta (UVT)
Mr Walid Chebbi (UBCI)

Année Universitaire : 2014 / 2015

Remerciements

Je remercie tous ceux qui mon aidé, de près ou de loin, dans l'élaboration et l'achèvement de ce travail dans le cadre du projet de fin d'étude.

Ma reconnaissance s'adresse plus particulièrement à :Mon encadreur universitaire ***Mme Hela Boucetta*** *et mon encadreur industriel* ***Mr Walid Chebbi*** *direction de système d'information de l'UBCI pour l'aide précieuse qu'ils m'ont apporté et pour tous les conseils judicieux qu'ils m'ont prodigué*

Enfin je remercie tous les enseignants de l'UVT pour leurs formations d'adéquation durant les cours universitaire

Merci

Dédicaces

Je dédiée ce travail à mes parents mon fils mon frère Amine mes sœurs et ma femme qui m'a soutenu et m'a encouragé tout au long de mes études Que Dieux leurs réserve bonne santé et longue vie.

Une pensée pour tous ceux qui m'ont soutenue de près ou de loin durant mes études.

Merci

SOMMAIRE

Table des Figures

Introduction Générale

Les réseaux de transmission de données ne cessent de s'accroitre, les volumes de données échangés augmentent de plus en plus. Tout est désormais informatisé de nos jours, et que le domaine de la technologie d'information(IT) devient prédominant. Les entreprises mettent en place des infrastructures réseau et systèmes plus ou moins complexes pour garantir un service fiable et un accès à l'information quasi permanent et "Pour que son infrastructure informatique lui donne entière satisfaction, toute entreprise doit pouvoir compter sur un réseau haute performance [wp1]. Ainsi, pour atteindre ces objectifs, tout administrateur réseau doit mettre en place des outils nécessaires et suivre des procédures standard de gestion de réseau.

Le gestionnaire de réseau doit donc surveiller en permanence les nœuds réseau, une perte de connexion à un nœud engendre une perte de temps et un risque opérationnel et financier. Sa tâche principale et d'assurer la surveillance au quotidien du comportement du réseau de l'entreprise par la supervision des équipements qui le constitue, le suivi des états des liens réseau, la consommation de bande passante. Il doit aussi définir des procédures et des tableaux de bord de suivi, élaborer des rapports d'analyses réseau et de créer des sauvegardes de configurations relatives aux hôtes et équipements réseau. Et finalement, résoudre les éventuels incidents et pannes pouvant survenir.

Ceci dit, pour assurer une meilleure gestion de son infrastructure réseau, l'Union Bancaire pour le Commerce et l'Industrie (UBCI) doit disposer d'un outil de supervision fiable assurant les taches de surveillance réseau. Dans ce cadre ce présent rapport de fin d'études propose en premier lieu l'étude de l'existant à l'UBCI .Cette étude nous permettra de dégager une problématique et en proposer une solution adéquate. Les besoins fonctionnels seront détaillés dans un second lieu.

Par la suite, on abordera la phase la plus importante qui est la conception et on finira par le développement de l'application et le test de son bon fonctionnement.

- L'étude, la critique de l'existant et la solution proposée.
- Spécification de besoins et étude théorique.
- La réalisation et test de fonctionnement

Chapitre 1 :

Cadre Général du projet

Dans ce chapitre nous allons présenter en premier lieu le projet dans son cadre général et les objectifs pour lesquels il a été conçu et développé. Nous allons par la suite décrire l'état de l'art ainsi que la problématique liée aux pannes et leur ampleur, la limite des solutions existantes. Enfin, nous présenteront le travail à réaliser, la méthodologie de conception adoptée ainsi que l'environnement.

1. Présentation de l'organisme d'Accueil :

A-Présentation de l'UBCI :

L'Union Bancaire pour le Commerce et l'Industrie, est une banque commerciale tunisienne fondée en décembre 1961, elle est fondée à la suite de la fusion de la Banque nationale pour le commerce et l'industrie-Afrique **(BNCI-A)** et l'Union financière et technique de Tunisie **(UFITEC)**.

Détenant plus de 50% du capital de l'UBCI, le groupe bancaire français BNP Paribas lui apporte son savoir-faire dans le métier de la banque, dans les nouvelles technologies de l'information ainsi que l'appui de son réseau international réparti sur 85 pays à travers le monde [ub].

L'UBCI a développé toute une gamme de produits et services pour ces clients particuliers afin de couvrir leurs besoins quotidiens, leurs financements de projets, la gestion de leur épargne, la couverture financière des familles en cas d'accident ou de décès et la préparation de leur retraite. Elle occupe également une place prépondérante dans le domaine du financement des nouvelles technologies. En outre, elle offre un ensemble de présentations à ses clients professionnels, dans les domaines des activités de marché, de banque d'affaires, de commerce international et d'ingénierie financière, et permet un accès direct au réseau mondial du groupe BNP Paribas.

B-Direction des Systèmes d'Informations DSI :

Au sein de l'**UBCI**, les systèmes d'information et les réseaux informatiques sont gérés par la direction des systèmes d'Information **« DSI »**.

Rattachée directement au Secrétariat Général, la **DSI** a pour mission de :

- Piloter les infrastructures informatiques.
- Administrer les serveurs d'application.
- Assurer la production et les traitements informatiques.
- Gérer les réseaux informatiques

Veiller à la sécurité et à la continuité des activités névralgiques de la banque.

La **DSI** est composée d'un effectif pluridisciplinaire de 27 collaborateurs composé de :

- Administrateurs de base de données.
- Administrateurs des systèmes Unix et Windows.
- Ingénieurs des réseaux informatiques et Telecom
- Développeurs informatiques.

Des équipes techniques pour l'exploitation informatique.

La banque comprend deux Centres de traitement de données (Data Centers) dont un principal de production et sis à Place Pasteur et un second utilisé comme centre de repli (Disaster Recovery Site (DRS)) au Charguia qui sont sous la responsabilité de la DSI.

2. Etude de l'existant

L'UBCI a mis en œuvre une solution de Communications Unifiées depuis 2009 en remplacement de l'infrastructure téléphonique traditionnelle par la technologie Téléphonie sur IP. Ce projet a nécessité la migration de ces liens WAN Frame Relay vers la technologie MPLS.

Ce réseau de plus d'une centaine de sites (dont 111 agences sur le tout le territoire tunisien) est un réseau totalement maillé MPLS de TUNISIE TELECOM avec des backups en Faisceaux

Hertziens (pour les sites importants) d'Orange Tunisie ou en « ADSL Data » (pour les petites agences).

Les sites centraux de Liberté Mégrine et Charguia disposaient de routeurs redondants (HSRP) pour assurer la haute disponibilité de l'adresse IP LAN des routeurs. Le côté propriétaire de HSRP n'était pas gênant puisque tous les routeurs étaient de marque Cisco.

Le reste des sites disposent chacun d'un seul routeur **Cisco 2811** avec deux connexions **WAN** redondantes (**MPLS L3** et **ADSL data** pour les petits sites et **FH** pour les sites Importants) et dont l'architecture logique peut être schématisée par la figure 1 :

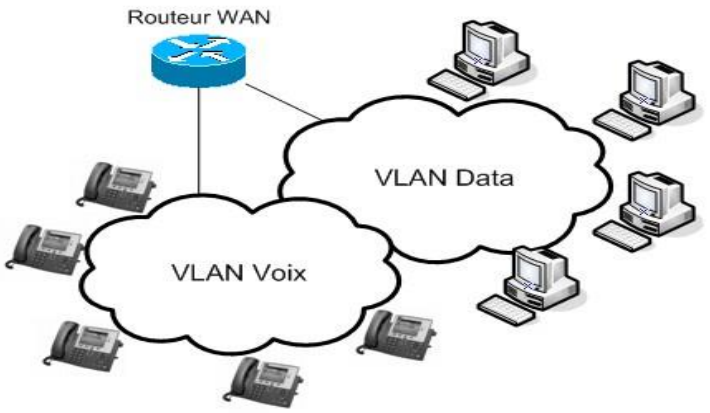


Figure 1 Architecture logique du réseau LAN

Les flux voix et les flux data sont séparés sur deux **VLANs** dans chaque site (cf fig1). Et que l'accès aux ressources voix (serveurs Cisco Call Manager) et données (Serveurs Métiers, Web, application..) est centralisé au niveau du site central à Tunis Liberté.

La figure 2 présente l'architecture du réseau domestique de la banque et le schéma d'interconnexion des sites à travers les liens de deux opérateurs Data.

Pour les connexions Backup de Tunisie Télécom les accès ADSL Data acheminent le trafic depuis la CE du client, la connexion aboutit sur un DSLAM, puis un routeur, puis un PE du nuage MPLS. L'ADSL est vu comme une technologie de connexion alternative au nuage MPLS (avec un support physique alternatif).

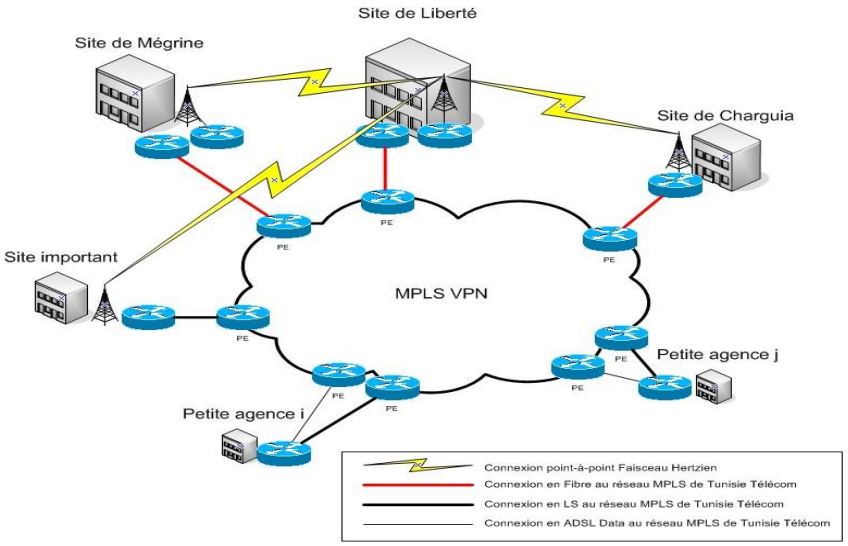


Figure 2 Topologie du réseau WAN de l'UBCI

Etant donné, que l'activité bancaire est au cœur de l'activité économique de tout pays et que l'image de toute banque repose sur le service qu'elle fournit, il est indispensable d'assurer la continuité de l'activité informatique. L'UBCI a donc mis en place une solution de supervision des accès WAN connu sous le nom de WATCH et une autre solution de monitoring des équipements LAN (switches), **Cisco Works.**

3. Problématique

Malgré le déploiement des deux outils WATCH et Cisco Works, les objectifs souhaités n'ont pas été atteint. En effet l'outil WATCH se base sur des pings (cf. fig3). Il s'agit d'un outil très basique qui se base sur des commandes ICMP. Alors que l'objectif des administrateurs réseau de la banque est disposé d'un outil complet qui supervise les liens, leur occupation de bande passante, les changements des états dans le routage dynamique.

Quant à l'outil Cisco Works, il est destiné aux équipements de marque Cisco et notamment dans l'archivage des fichiers de configurations et les images de systèmes d' exploitation des switches et routeurs. L'affichage des alertes et évènements ne permet pas une lecture aisée. En plus cet outil

n'est pas en mesure de donner des informations sur l'utilisation de la bande passante (Bandwith IN/OUT). (cf fig4)

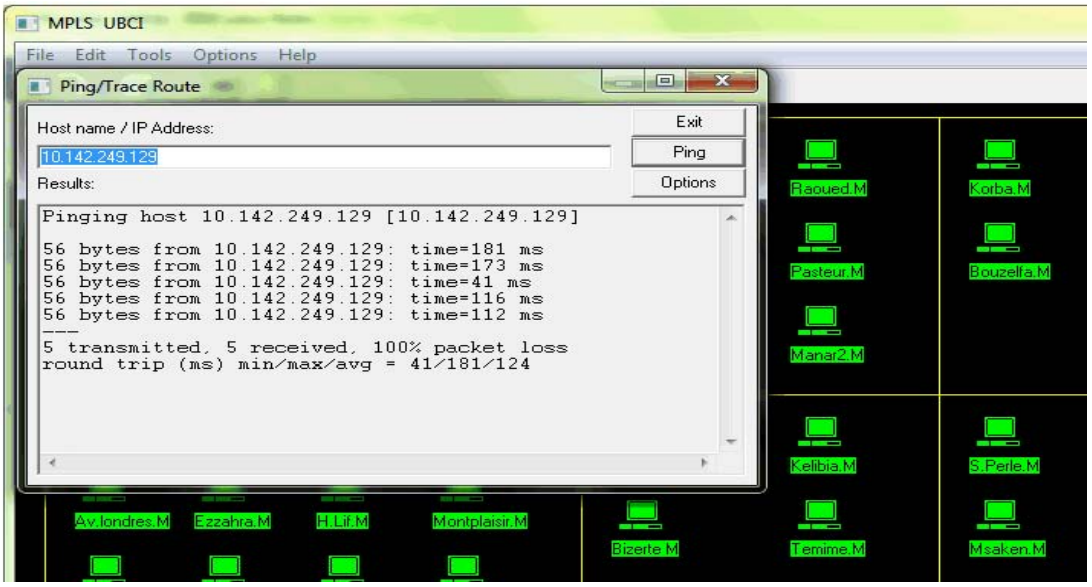


Figure 3 Interface graphique de supervision des accès MPLS de la banque

Un autre limite de l'outil Cisco Works est qu'il est inapproprié pour une opération de supervision standard d'un réseau hétérogène d' équipements Vu que la parc de la banque comprend des équipement d'autre constructeurs tel que Riverbed et Juniper .

Figure 4 Interface principale Cisco LMS

Nous pouvons conclure que, malgré l'existence de deux outils présumés de supervision, les objectifs n'ont pas été atteints. La banque tend toujours à disposer d'une application globale fiable et consistante.

4. Solution proposée

Pour répondre aux besoins de l'UBCI on se propose de développer un outil simple consistant et fiable permettant la gestion et la supervision des serveurs, routeurs, et autres équipements constituant son réseau.

Cet outil devra assurer en premier lieu la surveillance de ces équipements et ressources réseaux (bande passante), et en un second lieu, permettra de détecter de façon rapide les pannes pouvant affecter ces équipements. Tout évènement déclenché par un nœud quelconque du réseau devra être remonté à l'application centrale et enregistré dans une base de données dédiée.

La consultation en temps réel ou en différé devra être assuré à travers une interface WEB qui permet de donner une vue d'ensemble du réseau et des pannes pouvant surgir .

Conclusion

Dans ce chapitre nous avons situé le projet dans son cadre général. Nous avons présenté l'architecture réseau existante de la banque et détaillée les solutions existantes montré leur limites et proposé de développer une solution.

Dans le chapitre suivant, nous allons spécifier les différents types de besoins aussi bien que les acteurs et les architectures.

Chapitre 2 :

Etude théorique

Dans ce chapitre, nous allons effectuer une étude théorique des protocoles de supervision standard notamment SNMP qui adopté par la banque l'UBCI. Nous ferons également une analyse de son fonctionnement qui nous sera utile ultérieurement dans la spécification des besoins et la conception de notre solution.

1. Protocole de supervision

Il existe une panoplie de protocoles et d'outils de supervision de réseau aidant à collecter des informations sur les nœuds réseau et vérifier leur bon fonctionnement tels que SSH, ICMP, et SNMP. Mais en termes d'efficacité et consistance c'est le protocole SNMP qui se situe en premier rang.

a. le protocole SNMP

Comme son nom l'indique, le protocole SNMP ou Simple Network Management Protocol est un protocole C'est un protocole de la couche 7 du modèle OSI (couche application) qui permet de gérer tout équipements ou poste de travail pouvant être connecté au réseau.

Il est incontournable dans l'analyse des informations et évènements de tous les types (CPU , Température réseaux, systèmes….) et la détection des éventuelle pannes qui peuvent surgir . Ce fameux protocole est supporté par pratiquement tous les constructeurs de matériel informatique et réseau (Microsoft , Cisco, HP , Juniper …) et configurable sur la majorités des plateforme) Malgré les recommandation sécuritaires de le désactiver .

Ce protocole est donc utilisé pour effectuer plusieurs fonctions, qu'on peut citer :

- Equipements réseaux : Etat des interface (Up/Down) , Débit , charge de la ligne

- Périphériques d'impression : Etat consommables, charge tiroir …
- Onduleurs
- Caméras IP de surveillance
- Capteurs de température et de poussière

a.1. Les versions

Depuis son premier développement au sein de l'IETF (Internet Engineering Task Force, voir RFC 1157 Annexe1) au début des années 90, le protocole SNMP a connu plusieurs améliorations visant à optimiser son utilisation dans la supervision des réseaux en passant par plusieurs versions :

➢ *SNMPv1*

C'est la première version de ce protocole. La console de supervision interroge l'agent SNMP par un datagramme UDP sur le port 161. Cet UDP contient : la version de l'SNMP (0 pour SNMPv1), le nom de communauté déterminant les droits d'accès, la requête (get-request, get-next-request) et l'objet Identifier (OID,voir Annexe 1). La réponse de l'agent contient un datagramme contenant la requête get-response, avec pour et la valeur demandée correspondant à l'OID et un code d'erreur. L'agent peut également être configuré pour renvoyer des alertes à la même console par un datagramme sur le port 162 contenant la requête dite Trap.

Mais cette version présente quelques limites telles

- Absence d'un mécanisme adéquat pour assurer la confidentialité et la sécurité des fonctions de gestion.
- Faiblesse dans l'authentification et le cryptage, et facilité

➢ *SNMPv2*

Il s'agit d'une évolution de l'ancienne version, par l'introduction de quelques nouveautés tel est l'exemple de l'opération GETBULK, qui permet à une plateforme de gestion, de demander en bloc de plusieurs variables consécutives dans la MIB de l'agent.

Généralement, on demande autant de variables que l'on peut mettre dans un paquet SNMP. Ceci règle un problème majeur de performance dans SNMPv1. Avec la version 1, la plateforme est obligée de faire un GETNEXT et d'attendre la réponse pour chaque variable de gestion,

➢ *SNMPv3 :*

Cette nouvelle version du protocole SNMP a été développée pour renforcer le volet sécurité des transactions. Ce volet comprend l'identification des parties communicantes et le cryptage de leur conversation.

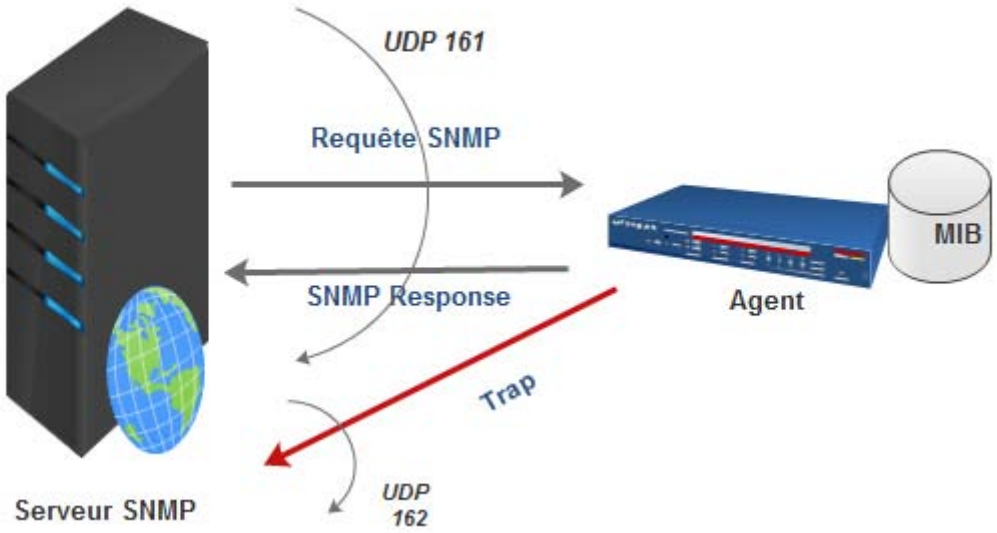


Figure 5 Mode de fonctionnement de l'SNMP

b- Communautés

SNMP fonctionne par des groupes d'agents ou communautés .Du côté de l'agent ont créé une communauté dite publique accessible pour tous mais en lecture seule et deuxième une communauté privée qu'on peut affecter un nom au choix et accessible en lecture/écriture et protégée par un mot de passe.

Du côté serveur, on ajoute le ou les hôtes avec les OID de la MIB nécessaires à la supervision.

c- La MIB

c.1. Présentation

Pour que SNMP fonctionne, il est nécessaire qu'un protocole d'échange soit définit. Il y a aussi une standardisation des informations que ce protocole peut transporter. C'est un protocole Internet, il doit être utilisable sur des plates-formes hétérogènes (matériel comme système d'exploitation).

Pour cela qu'on doit parler de MIB (Management Information Base). En fait, la MIB est une base de données des informations de gestion maintenue par l'agent. C'est la base à laquelle on va demander les informations.

c.2. Structure de la MIB

La MIB a une structure hiérarchique, les informations sont regroupés sous la forme d'un arbre, chaque information a un identificateur OID qui est représenté par une suite de chiffre séparé par des points, ceux-ci l'identifie d'une façon unique, et un nom indiqué dans le

Document qui décrit la MIB, pour être lisible pour un utilisateur.

La figure suivante décrit Une partie commune à tous les équipements d'un même type dans la MIB.

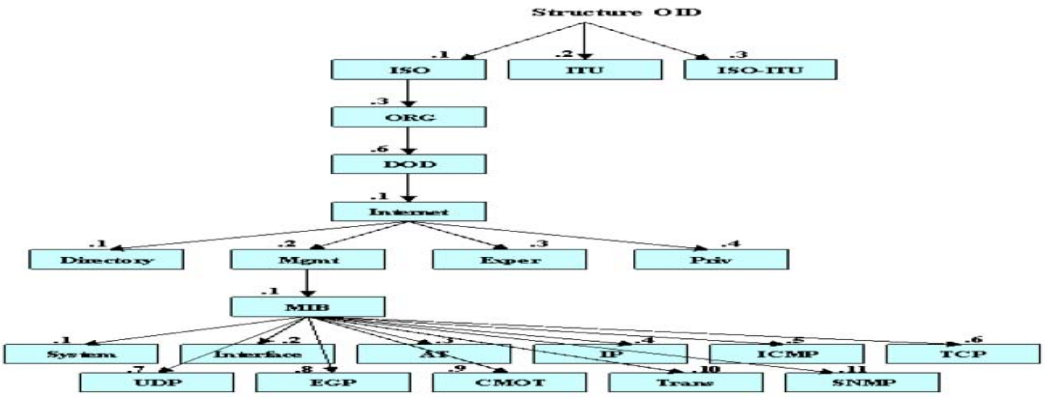


Figure 6 Structure de la MIB

Conclusion

Dans ce chapitre nous avons effectué une étude théorique sur l'opération de supervision réseau ainsi que l'analyse des différents protocoles de supervision notamment le protocole SNMP qu'on a étudié avec plus de détail ce qui nous aidera ultérieurement dans les phases de conception et réalisation.

Chapitre 3 :

Spécification des Besoins

Dans ce chapitre, nous allons spécifier les besoins fonctionnels et non fonctionnels de l'application, ce qui nous amènera à identifier les possibilités du système et les besoins des utilisateurs que nous essayerons de les projeter dans des diagrammes de cas d'utilisations globales et détaillés.

1. Spécification des Besoins

La solution cible doit satisfaire les besoins fonctionnels qui seront exécutés par le système et les besoins non fonctionnels qui identifient la qualité logicielle du système.

1.1. Les Besoins Fonctionnels

Cette application doit couvrir principalement les besoins fonctionnels suivants :

- ✓ Authentification
- ✓ Gestion des utilisateurs : ajout et suppression. Cette fonction est possible pour l'administrateur seulement et masqué pour les autres utilisateurs.
- ✓ Cartographie des nœuds réseaux et suivi en temps réel des équipements. Ce qui permettra aux différents acteurs et selon leur rôle de surveiller l'état des chacun des équipements (up/down, idle / faulty …)
- ✓ La gestion des logs et archivage des évènements dans une base de données pour d'éventuelles consultations
- ✓ Suivi des charges sur les liens WAN (Occupation de bande passante)
- ✓ La possibilité d'avoir un accès à l'application à travers un agent Android

2.2. Les Besoins Non-Fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Les principaux besoins non fonctionnels de notre application ce résument dans les points suivants :

- ✓ l'application doit être portable c'est-à-dire sa capacité à pouvoir être adapté plus ou moins facilement en vue de fonctionner dans différents environnements d'exécution.
- ✓ Le code doit être clair pour permettre des futures évolutions ou améliorations
- ✓ L'ergonomie : l'application doit offrir une interface conviviale et facile à utiliser

2.3. Les Besoins Architecturaux

L'application doit fonctionner dans un réseau. Certes l'accès doit donc être possible à partir de n'importe quel poste par le biais d'un navigateur web .Mais , l'importance d'avoir un système d'archivage de donnée (parmi le besoins fonctionnels) et de gestion de logs nous oblige de créer une base de données permettant de stocker les différents évènements .

Ainsi l'architecture à trois niveaux s'avère la mieux à adopter (aussi appelée architecture 3-tiers) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

En effet, l'architecture 3-tiers, ou encore architecture à trois niveaux ou couches est l'extension du modèle client –serveur. L'architecture logique du système est divisée en trois niveaux ou couches :

- couche présentation ;
- couche métier ;
- couche accès aux données.

Comme l'indique la figure 5, le client (couche présentation) accèdent au serveur pour demander une ressource. Ce dernier (couche métier) fait appel au serveur secondaire (généralement un serveur de base de données) pour livrer les données demandé aux clients.

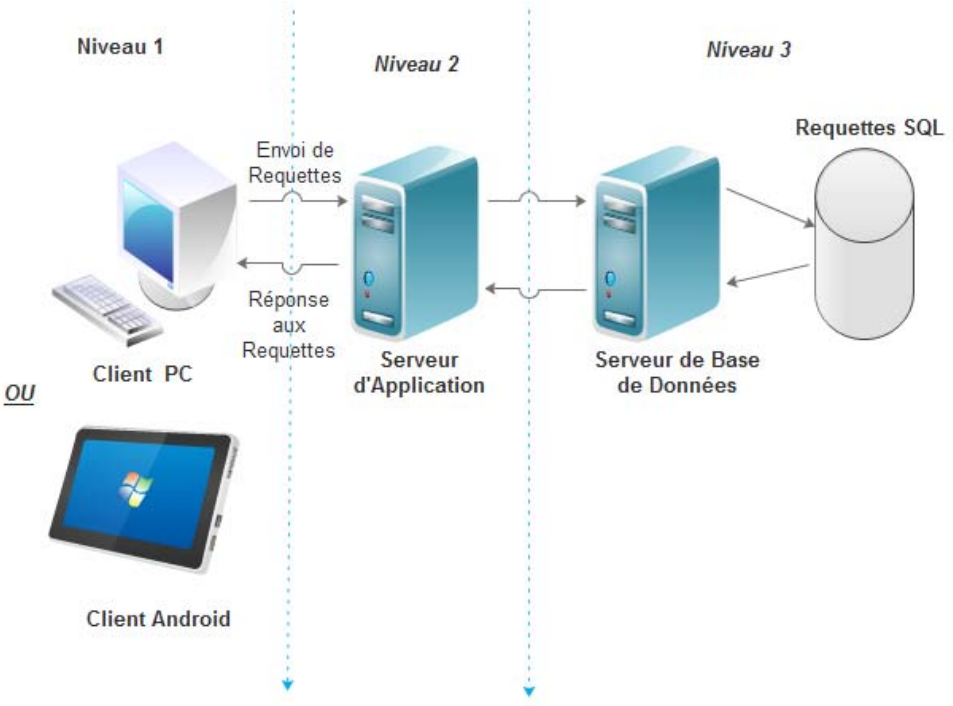


Figure 7 Architecture 3 –tiers adoptée

3. Modélisation des Besoins

Nous avons choisi le langage de modélisation UML pour présenter les diagrammes des cas d'utilisation ainsi que les acteurs de notre application.

3.1. Pourquoi UML ?

UML ou encore Unified Modeling Langage, est un langage de modélisation des données et des traitements, formel et normalisé qui offre un standard de modélisation. C'est alors un langage graphique qui entre dans l'optique de la programmation orientée objet et qui sponsorisé par les grands calibres du monde de l'informatique tel que HP, Microsoft et IBM.

Vu que l'interface de supervision principale est une interface web, les diagrammes de classes obtenus doivent être encore modifiés selon la technologie utilisée (J2EE, PHP….).

Nous utiliserons dans notre modélisation le logiciel ***StarUML*** qui est un des logiciels de modélisation UML le plus populaire. Simple et facile à utiliser, son éditeur l'a rendu sous une licence modifiée de GNU CPL.

3.2. Les Diagrammes De cas d'utilisation

Le digramme Uses Cases montre les interactions fonctionnelles entre les acteurs et le système à l'étude

*Acteur :

Rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.

*Cas d'utilisation (use case) :

Ensemble de séquences d'actions réalisées par le système produisant un résultat observable intéressant pour un acteur particulier. Collection de scénarios reliés par un objectif utilisateur commun.

*Association :

Utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement « participe à ».

*Inclusion :

Le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements.

*Extension :

Le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension

*Généralisation :

Les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des relations spécifiques supplémentaires avec d'autres acteurs ou cas d'utilisation.

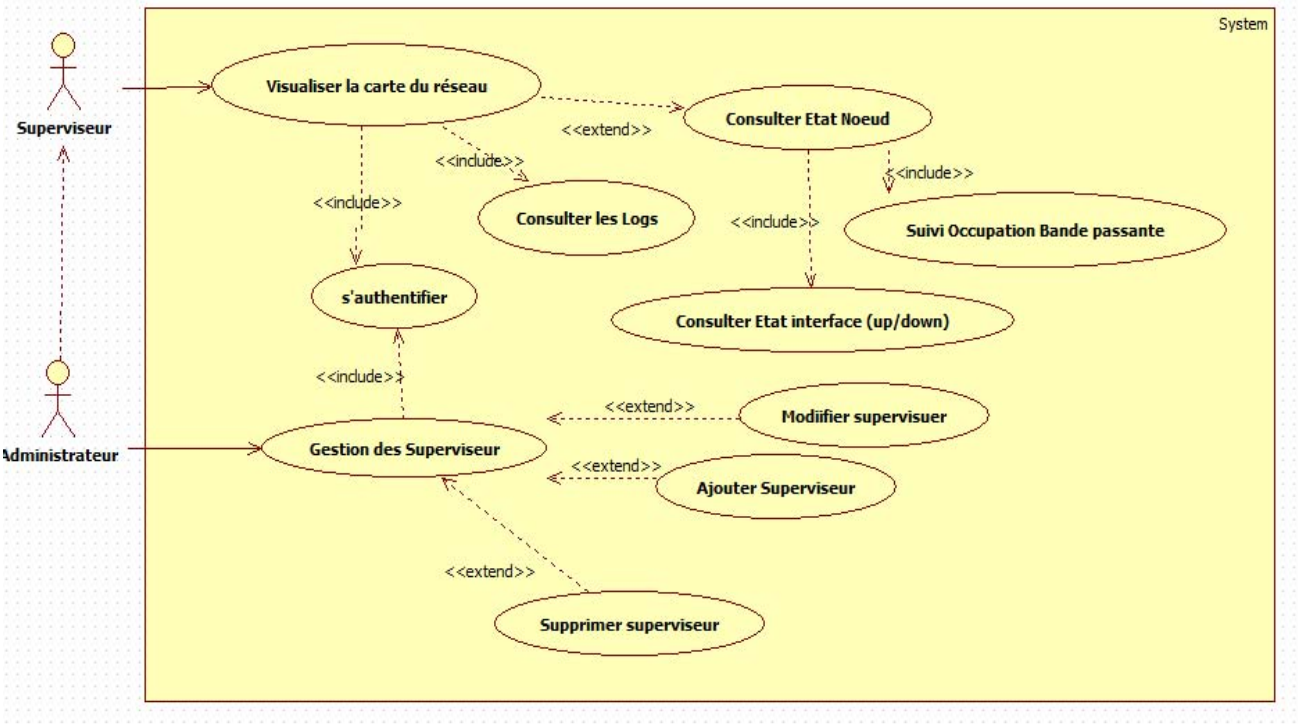


Figure 8 Cas d'utilisation Général

*Description textuelle du cas d'utilisation Général

- Pour accéder au système il faut obligatoirement saisir un login et un mot de passe quel que soit l'utilisateur.
- L'acteur administrateur (super superviseur) a deux volets sur le système :
  - ✓ La gestion des utilisateurs qui permet d'ajouter, modifier ou bien supprimer utilisateur (Superviseur).
  - ✓ La gestion et la visualisation de la cartographie du réseau
- L'acteur superviseur a pour mission de consulter uniquement l'état du système sans avoir l'habilitation d'effectuer des modifications.

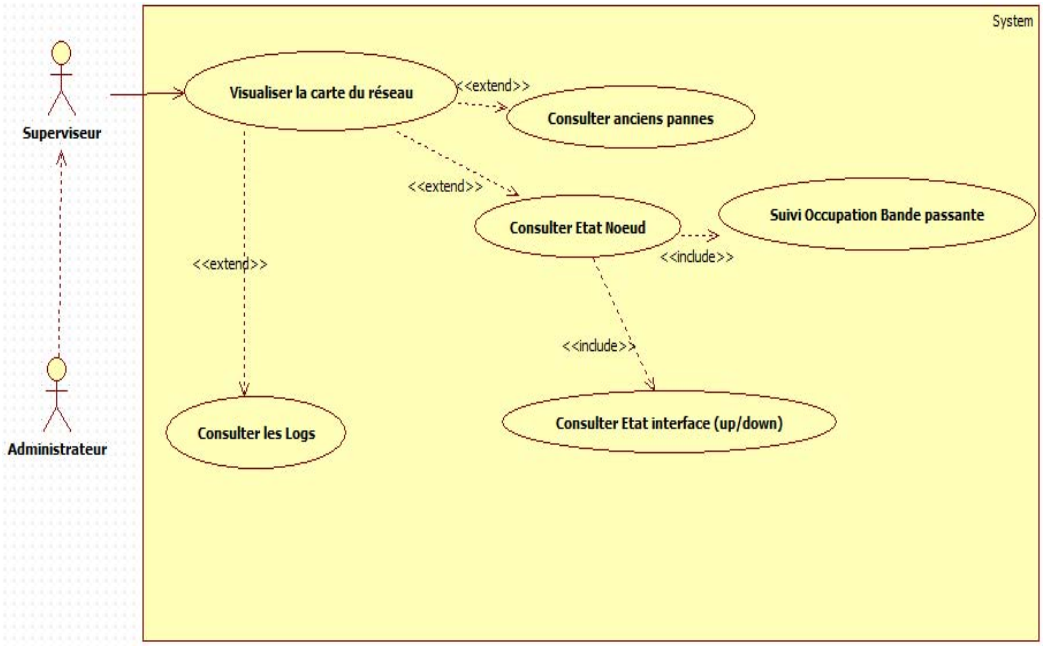


Figure 9 Cas d'utilisation « Visualisation de la carte du réseau »

*Description textuelle du cas d'utilisation Visualisation de la carte du réseau

La consultation d'état de système : consulter état serveur, consulter état routeur, consulter panne… est la même pour tous auteurs, elle est précédé par l'authentification.

*Description textuelle du cas d'utilisation Gestion des comptes

Seul l'administrateur réseau (super superviseur) peut créer, modifier, ajouter, et supprimer un compte et un mot de passe. Les comptes et les mots passe valides sont initialisées et enregistrées par l'administrateur réseau (super superviseur), elle est précédé par l'authentification.

Conclusion

Dans ce chapitre nous avons étudié des différents besoins fonctionnels défini les acteurs suivant on va entamer la phase de conception de système à réaliser ainsi que l'architecture de notre application.

Chapitre 4 :

Conception

Ce chapitre est une description de la phase de conception qui est peut être considérée comme étant la clé de succès du projet .Nous allons décrire les différentes étapes de conception de la partie statique, les données, et la partie dynamique ainsi que les traitements. Soit donc définir précisément chaque sous-ensemble du logiciel.

1. Conception Architecturale

La conception de l'architecture fonctionnelle donne une idée sur la logique de fonctionnement du système, les services souhaités et les performances attendus.

1.1 Architecture de l'application

L'architecture cible de notre solution peut être schématisée dans la figure suivante.

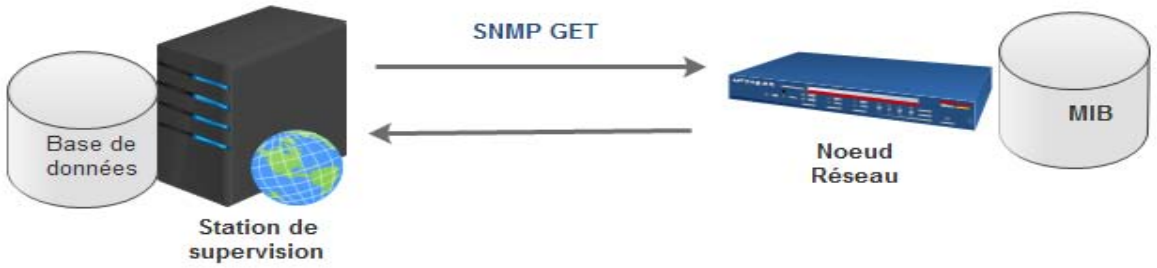


Figure 10 Schéma d'architecture cible

Le code principal de l'application devrait contenir des petits codes ou encore micro codes assurant la collecte des informations sur l'état des nœuds réseau et retourner ces états en langage compréhensible. Le programme devrait tourner en boucle continu, et renvoyer les évènements reçus à une base de données.

L'interface WEB servira d'interface avec le système de supervision et permettra à l'utilisateur de visualiser les états des différents équipements avec les dates et heures de déclenchement des différents évènements.

Ainsi, l'architecture logique d'un système est l'architecture applicative voire logicielle qui définit la répartition des tâches fonctionnelle. Elle est consacré à :

- Concevoir et structurer une application à partir de ses spécifications fonctionnelles (listé dans le chapitre II).
- Décomposer de façon logique chaque partie de l'application en couches.
- Introduire les concepts de découpage en couches, modules, composants, design patterns et **Framework.**

L'architecte logique découpe le système suivant plusieurs vues.

Vue en couches :

Un vue « logique » montrant le découpage des fonctions de l'application, Elle est indépendante des considérations physiques. La décomposition logique de chaque application se fait en en 5 couches :

- Présentation
- Contrôleur
- Services
- Domaine
- Persistance

Chaque couche a son propre rôle et utilise la couche située en dessous d'elle.

Couche présentation :

La couche Présentation gère et assure l'affichage de l'interface graphique utilisateur ou les Interfaces Homme-Machine (IHM : fenêtres, pages …)

On distingue trois catégories d'IHM pour les applications interactives :

- ✓ Client léger : il ne nécessite qu'un navigateur Web coté client et aucun déploiement n'est réalisé sur le poste client, les différents écrans de l'application sont générés en temps réel côté serveur et téléchargés par le poste client
- ✓ Client lourd : l'ensemble des écrans de l'application sont stockés ou générés sur le poste client et doivent avoir été déployés sur celui-ci préalablement à l'exécution
- ✓ Client riche (client intelligent) : Ce modèle constitue un compromis entre le client léger et le client lourd, il présente une ergonomie comparable à celle d'un client lourd avec une limitation des déploiements.

Couche Contrôleur :

La couche Contrôleur gère :

- ➢ les droits d'accès
- ➢ le contrôle de la cinématique des écrans
- ➢ les erreurs et les exceptions qui peuvent être levées
- ➢ les espaces (sessions) de travail utilisateur

Couche Services :

La couche Services correspond aux traitements qu'effectue l'application elle représente l'implémentation de la logique des cas d'utilisation. Cette couche doit :

- ➢ implémenter le logique métier
- ➢ gérer la sécurité applicative
- ➢ gérer et vérifier l'intégrité des transactions

Couche Domaine :

La couche Domaine gère l'intégrité du modèle « métiers ». Cette couche intègre principalement:

- la gestion des règles métiers « élémentaires ».
- la fourniture des moyens d'accès aux données

La couche Domaine est concentrée sur le métier de l'entreprise, commun à toutes les applications

Couche Persistance :

La couche Persistance intègre principalement :

- la persistance complète du Système d'Informations.
- La fourniture des services de stockage des données, moteurs relationnels, bases objets …
- la création, la modification, la suppression d'occurrences des objets métiers

La couche Persistance offre les fonctionnalités de base qui permettent :

- de créer, rechercher, modifier et supprimer des composants objets métiers dans le respect des propriétés transactionnelles classiques
- d'utiliser le mécanisme de projection objet vers relationnel (mapping Objet / Relationnel) qui consiste en la transformation de la représentation des données en une représentation objet
- d'offrir le support, des contextes transactionnels issus de la couche domaine

Tout système d'information nécessite la réalisation de trois groupes de fonctions: le stockage des données, la logique applicative et la présentation. Ces trois parties sont indépendantes les unes des autres: on peut ainsi vouloir modifier la présentation sans modifier la logique applicative. La conception de chaque partie doit également être indépendante, toutefois la conception de la couche la plus basse est utilisée dans la couche d'au-dessus. Ainsi la conception de la logique applicative se base sur le modèle de données, alors que la conception de la présentation dépend de la logique applicative.

Le **Modèle Vue Contrôleur** (**MVC**) est une architecture et une méthode de conception pour le développement d'applications logicielles qui sépare le modèle de données, l'interface utilisateur et la logique de contrôle. Ce modèle d'architecture impose la séparation entre les données, les traitements et la présentation, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur.

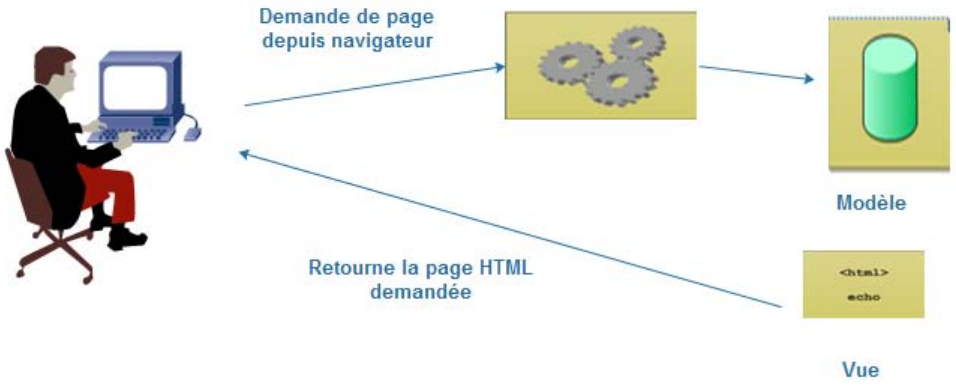


Figure 11 Modèle MVC

Architecture adoptée :

Dans notre projet nous allons adopter l'architecture qui est composé de 3 couches :

- Présentation
- Contrôleur
- Persistance

1.2. Conception de l'Architecture Physique

La conception de l'architecture physique élabore des solutions concrètes permettant d'exécuter l'architecture fonctionnelle du système.

L'architecture physique Est une structure de constituants (sous-systèmes et/ou composants technologiques) et de liens physiques qui les connectent ; ces éléments respectent les contraintes requises.

- concevoir l'architecture optimale d'un système complexe qui satisfait ses exigences techniques ;
- connaître les éléments qui composent une architecture fonctionnelle et une architecture physique ;
- savoir comment ces architectures sont obtenues et quelles sont leurs relations.

La vue en niveaux

La vue en niveaux (la tier view) donne une vision plus « physique » de la structuration de l'application. Les niveaux (ou tiers) peuvent être répartis physiquement sur différents composants matériels.

Des modèles standards de répartition de niveaux ont été définis dans les projets au fur et à mesure de l'évolution des capacités matérielles et des besoins

Dans notre cas on a une application web qui tourne dans un réseau local déployé sur un serveur http qui comporte aussi la partie donnée .l'application est accessible depuis des postes clients via un navigateur Web.

Déploiement logique/physique

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

La figure 11 donne une idée sur les différents mécanismes de déploiement

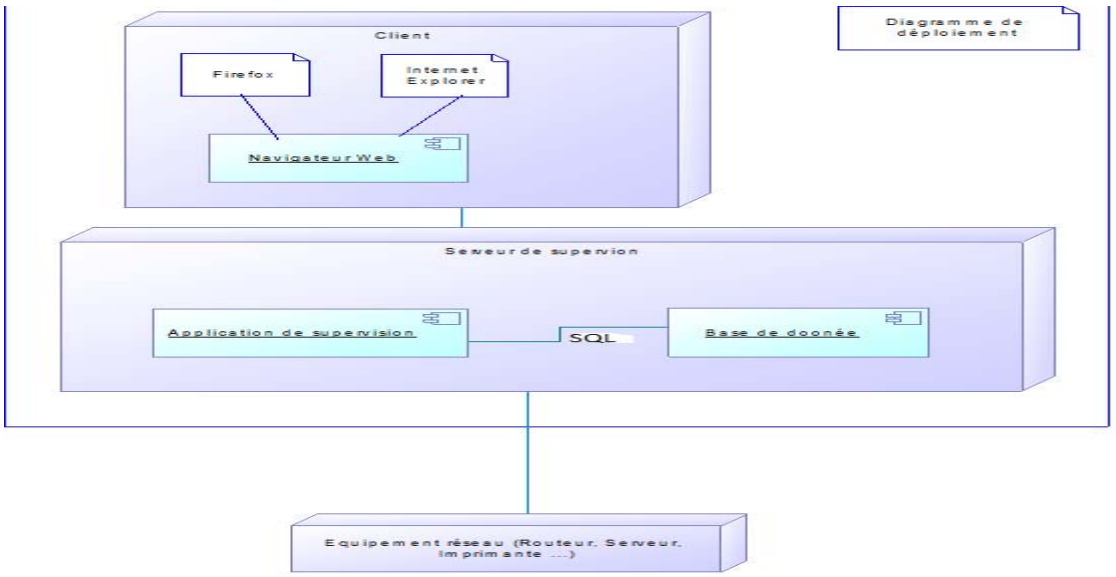


Figure 62 Diagramme de déploiement

2. Conception

La conception d'un logiciel est la mise en œuvre d'un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus permettent la conception, l'écriture et la mise au point d'un logiciel. Elle consiste à définir précisément chaque sous-ensemble du logiciel.

2.1. Conception de l'Aspect Statique

Cette étape permet de modéliser la structure logique du système, c'est-à-dire les aspects statiques du système. Cette modélisation est en grande partie effectuée dans des diagrammes de classes, avec éventuellement des diagrammes d'objets montrant des configurations spécifiques du système dans des conditions particulières. Le contenu principal de cette section est donc la présentation des éléments de modélisation du diagramme de classes.

2.1.1. Diagramme de Paquetage

Le diagramme de paquetages est une représentation graphique des relations existant entre les paquetages composant un système, dans le langage Unified Modeling Language (UML).

Le paquetage est uniquement un élément d'organisation et n'a pas de réalité concrète dans le système physique final

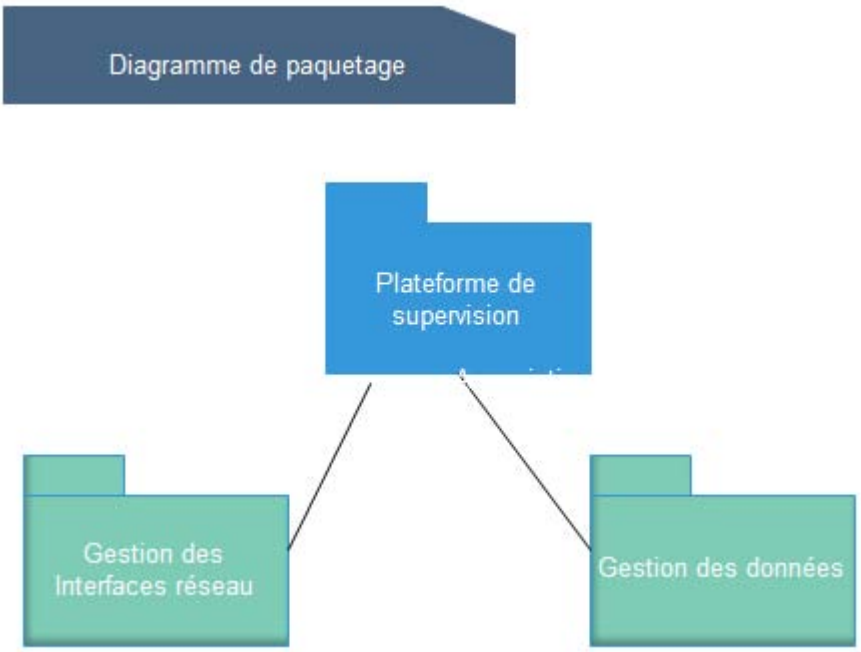


Figure 13 Diagramme de Paquetage

2.1.2. Diagramme de Classes

un diagramme de classes est une collections de modélisations statiques, c' est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.Les classes peuvent être liées entre elles grâce

au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un moule à objet : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

Le diagramme de classes donne une vue statique du système logiciel puisqu'il décrit les types et leurs objets de ce dernier. Typiquement, il met en relation des classes mais aussi des interfaces, des types de données, des types énumérés. C'est donc est un réseau statique de classes et d'associations. En partant des classes et des associations trouvées précédemment, il faut construire un schéma sous forme de représentation graphique dans lequel les classes seront représentées par des rectangles et les associations par des traits pleins. Il faut ajouter à ce schéma des informations concernant les classes et leurs associations. Le diagramme de classes simplifié est donné dans la figure suivante (Figure 9) Une classe est représentée par un rectangle séparée en trois parties :

- La première partie contient le nom de la classe
- La seconde contient les attributs de la classe
- La dernière contient les méthodes de la classe

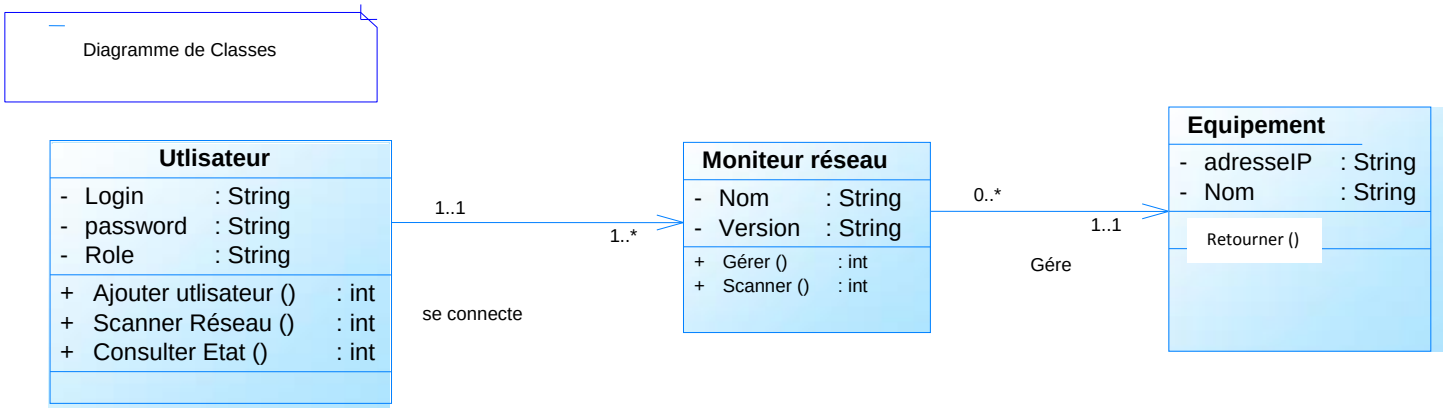


Figure 14 Diagramme de Classe

Et affiche à temps réel l'état des équipements, des ports, des routeurs… en se basent sur l'adresse du réseau propre à la banque 10.142.0.0 /16 qui est connu par chaque utilisateur autorisé à utiliser le programme qui sont généralement des administrateurs réseau et systèmes.

2.2. Conception Aspect Dynamique

- Points de départ : Modèles de la vue cas d'utilisation + diagrammes de classes
- Objectif : modéliser les algorithmes des cas d'utilisation

Après la construction des diagrammes de cas d'utilisation et d'activité, et la construction des diagrammes de classes et d'objets, la modélisation des aspects dynamiques répond globalement à la question « comment est spécifié le comportement du système, c'est-à-dire comment sont spécifiés les algorithmes des cas d'utilisation en parcourant le graphe de classes et des objets ? »

Le modèle dynamique montre donc le comportement du système et l'évolution des objets dans le temps. Il identifie les différents événements venant du monde externe et montre l'enchaînement

2.2.1. Diagramme d'activité

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallèles. Le diagramme d'activité est également utilisé pour décrire un flux de travail.

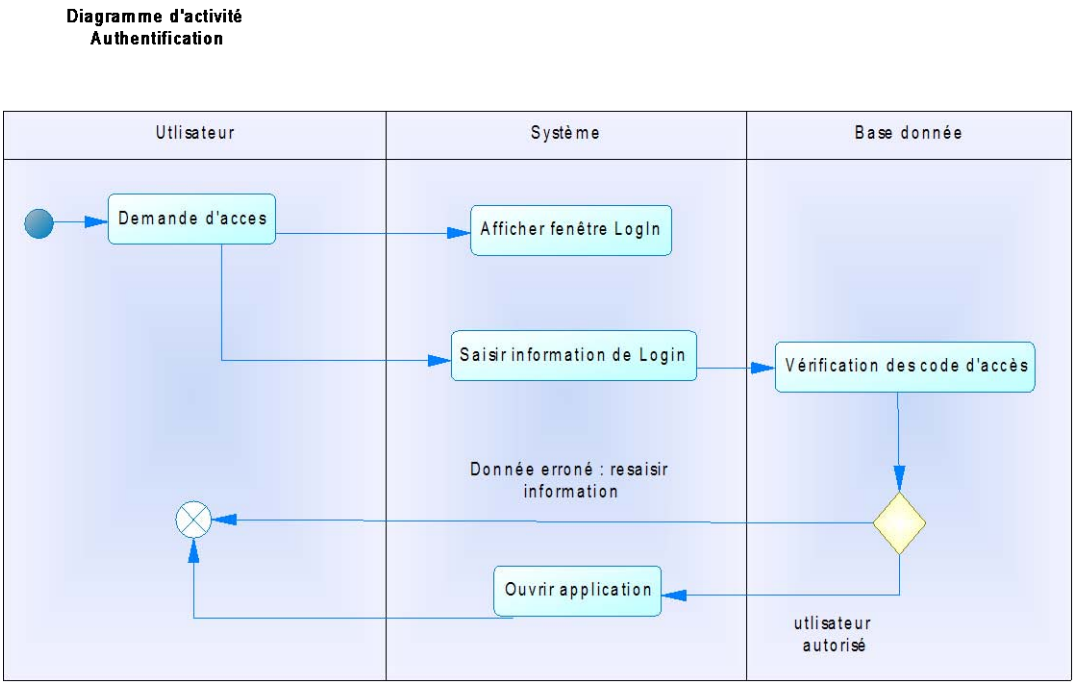


Figure 15 Diagramme d'activité Authentification

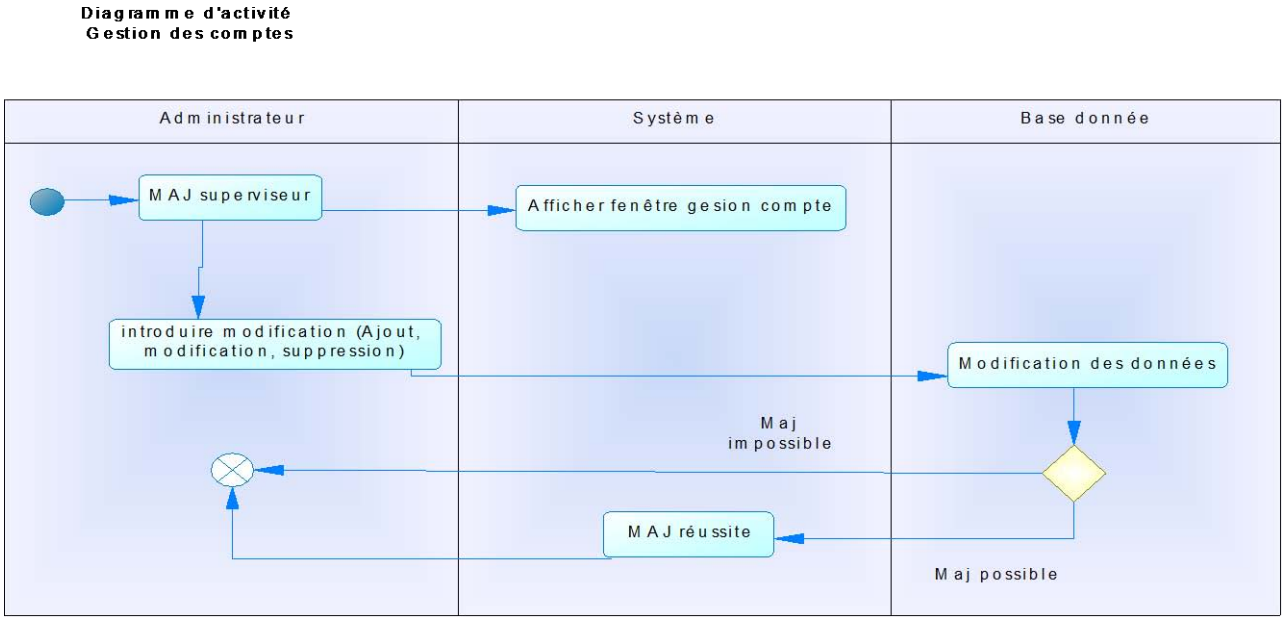


Figure 76 Diagramme d'activité gestion des comptes

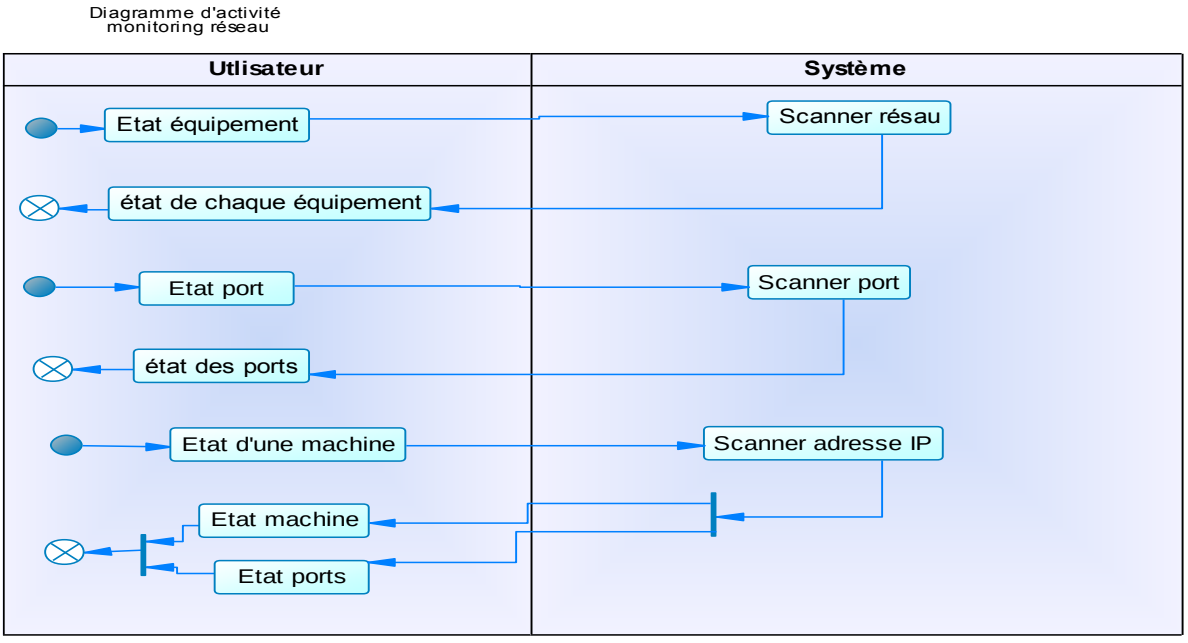


Figure 87 Diagramme d'activité gestion de réseau

2.2.2. Diagramme de Séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

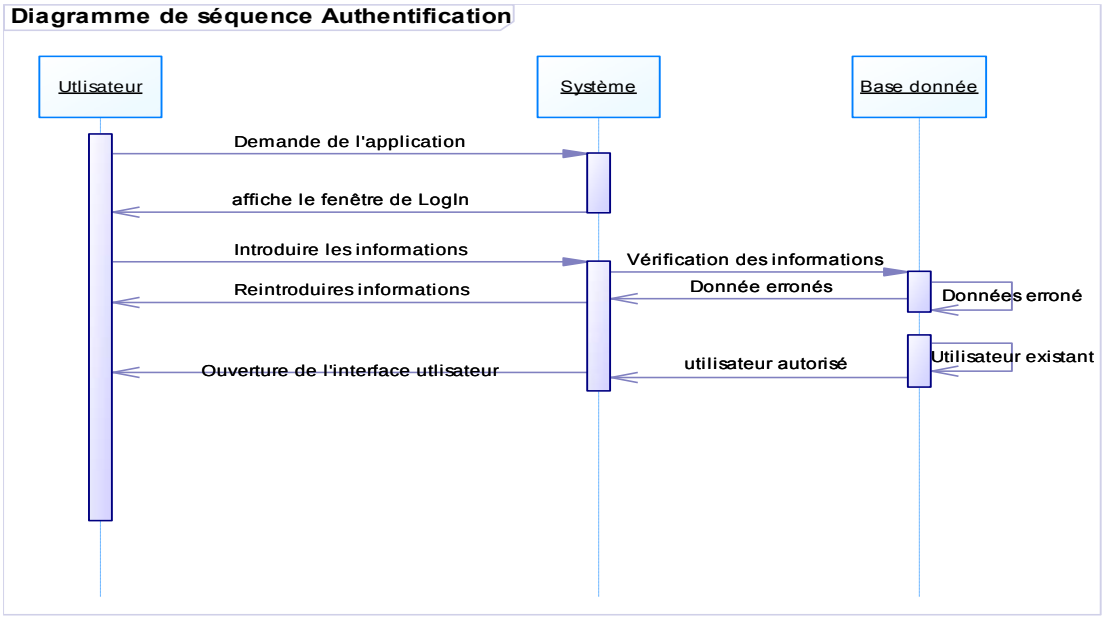


Figure 18 Diagramme de séquence Authentification

Description du scénario :

Le système affiche l'interface d'authentification.

- L'utilisateur introduit un login et un mot de passe.
- Le system vérifie le login et le mot de passe.
- Si les données saisies sont correct le système affiche l'interface de l'application, sinon le système demande de répéter la saisie de login et mot de passe

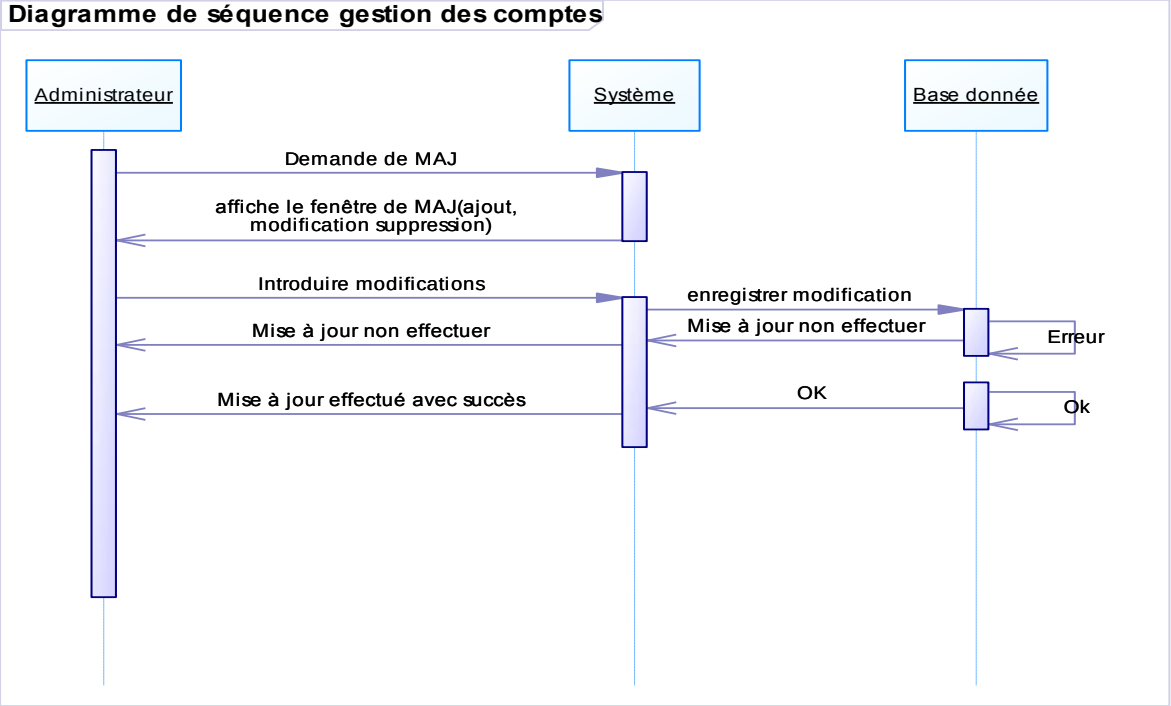


Figure 19 Diagramme de séquence gestion des comptes

Description du scénario :

Après l'authentification de l'administrateur le système affiche l'interface pour l'ajout, suppression ou la modification des coordonnées d'un utilisateur

- ✓ L'administrateur introduit les nouvelles notifications (les coordonnées d'un nouvel utilisateur ou la modification des coordonnées ou la suppression d'un utilisateur).
- ✓ Tous les mortifications sont enregistrés dans la base de donné

Diagramme de séquence monitor réseau

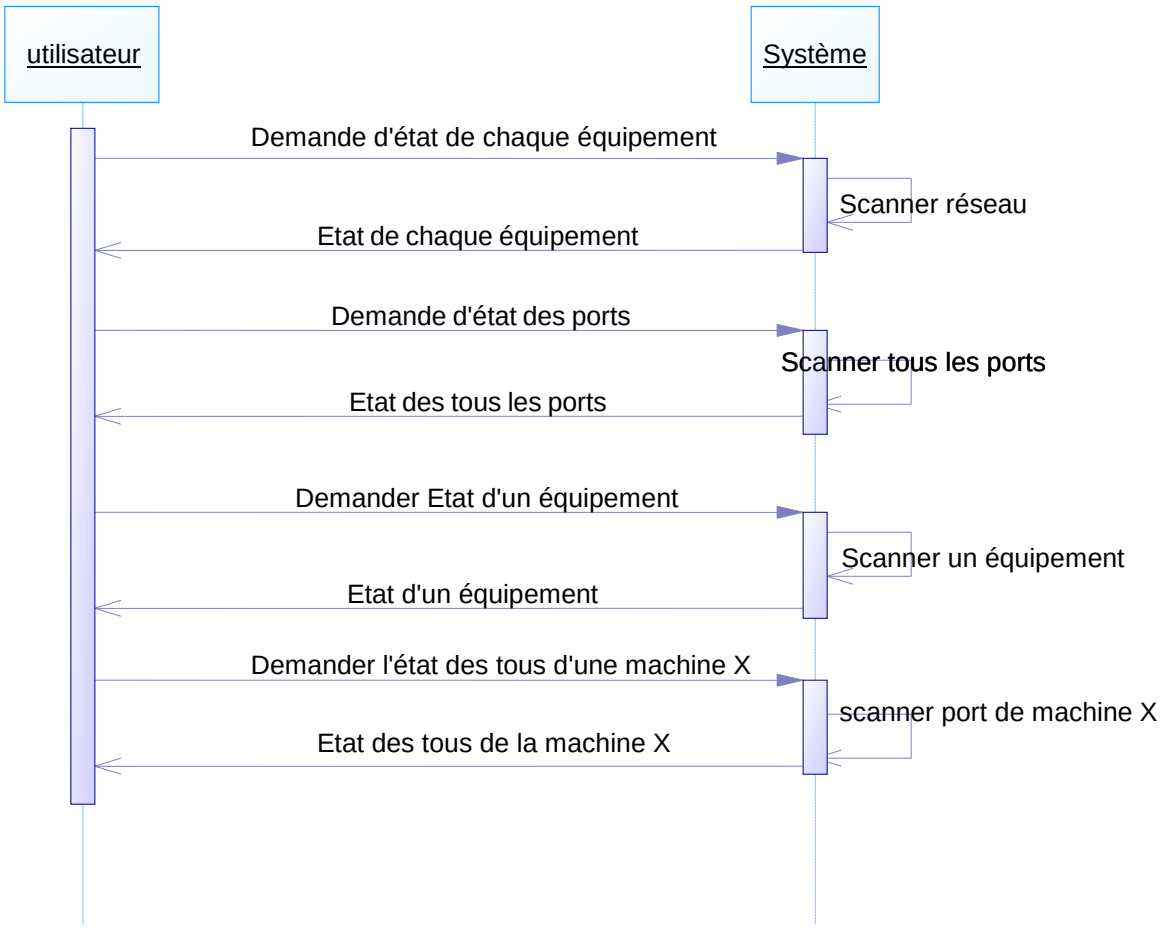


Figure 20 Diagramme de séquence gestion des comptes

Description du scénario :

- ✓ L'utilisateur demande au système l'état de chaque équipement (routeur, serveur et commutateur)
- ✓ Le système scanne le réseau
- ✓ L'utilisateur peut avoir l'état de chaque équipement
- ✓ L'utilisateur demande au système l'état des ports
- ✓ Le système scanne le réseau
- ✓ L'utilisateur peut avoir l'état de chaque port
- ✓ L'utilisateur demande au système l'état de performance des équipements
- ✓ Le système scanne le réseau
- ✓ L'utilisateur demande au système l'état d'une adresse IP
- ✓ Le système scanne le réseau

Conclusion

Dans ce chapitre nous avons achevé la partie la plus importante du travail qui est la conception détaillée avec ses différents aspects. Dans le chapitre suivant nous entamerons la partie réalisation.

Chapitre 5 :

Réalisation

Ce présent chapitre est une description de la phase de finition du projet. Il s'agit de présenter les différentes étapes de réalisation de l'application cible, l'environnement logiciel de développement les configurations nécessaire du protocole SNMP et quelques tests de bon fonctionnement. Nous donnerons quelques captures d'écran des actions effectuées.

1. Environnement logiciel

Tout le développement de l'application a été réalisé sur des machines dont le système d'exploitation Microsoft Windows 7 32 bits. L'environnement de développement utilisé se basait sur le langage de programmation PHP qu'on a associé avec le système de gestion de base de données MySQL server 5.5.20 et pour la programmation des requêtes SNMP nous avons utilisé le logiciel Net-SNMP.

1.1 Pépinière de développement WampServer

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL[wp2]. . Parmis les autres composantes de cette plate-forme , l'outil PHP MyAdmin qu'on utilisera dans la création et l'administration de notre base de donnée

WampServer dispose également de plusieurs fonctionnalités qu'on peut citer :

- Gestion des services Apache et MySQL
- Passage en mode online/offline
- Gestion des paramètres de configuration des serveurs
- Gestion des logs
- Accès aux fichiers de configuration

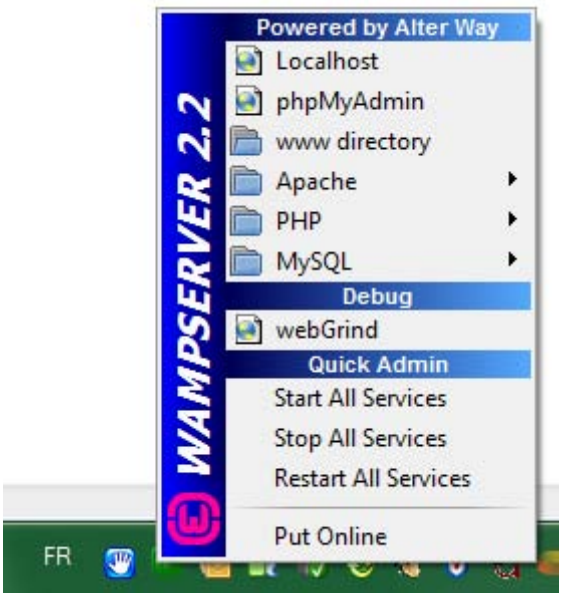


Figure 21 Fonctionnalités dans WampServer

1.2 Langage de programmation PHP

Le langage de programmation utilisé dans le développement de notre application sera le langage PHP (HyperText Preprocessor). Il s'agit est un langage utilisé principalement utilisé dans la création de sites web dynamiques et qu'il est souvent associé à des bases de données. Le serveur (qui généralement héberge le site web) interprète le code PHP et génère le code (constitué généralement d'HTML, de CSS ou de JavaScript) qui pourra être interprété par un navigateur(Client) .

PHP souvent installé sur un serveur Apache, et son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. .

1.3 Net-Snmp

Pour collecter les données sur les évènements déclenchés sur les nœuds réseau (switch ou routeur), il est nécessaire d'envoyer des requêtes SNMP de collecte. Pour utiliser ses requêtes dans PHP, il faut faire appel à des bibliothèques spécifique à l'utilisation du protocole SNMP .Un des solutions est d'installer l'outil Net-SNMP pour le support SNMP dans PHP.

En effet, Net-SNMP est un ensemble de logiciels permettant d'utiliser et de déployer le protocole SNMP (v1, v2c et v3) [wp3]. Ces logiciels contiennent une bibliothèque client générique, une suite d'applications en ligne de commande, un agent SNMP très extensible ainsi que des modules en Perl et en Python .

Une fois installé, il suffit d'activer l'extension php_snmp au niveau du volet PHP Extenions dans la plateforme WAMPServer (Comme le montre la figure ..)

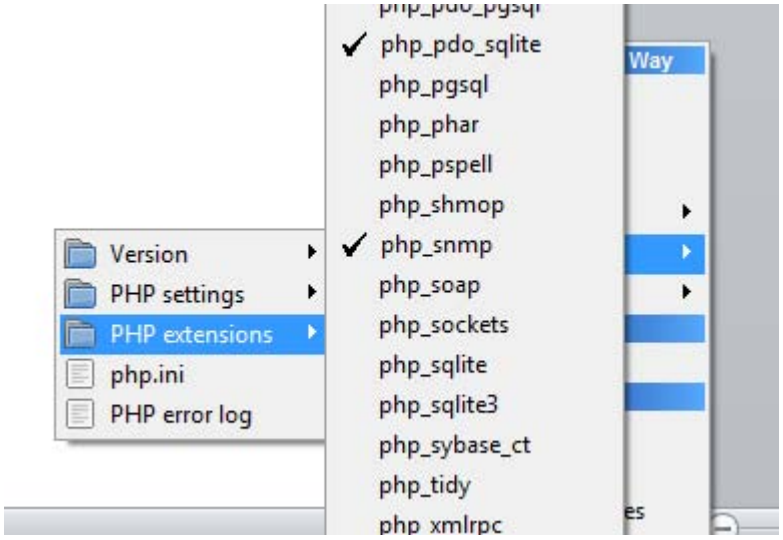


Figure 22 Ajout de l'extension php_snmp

1.4MySQL server 5.5.20

C'est un système de gestion de base de données (SGBD), dont l'objectif est d'obtenir des bonnes performances dans lecture. Il est distribué sous une double licence GPL et propriétaire. Il s'agit de l'SGBD le plus utilisé dans le monde en comparaison avec Oracle, et Microsoft SQL Server vu sa simplicité et son orientation vers le service de données déjà en place.

2. Environnement Matériel

1.1Activation de l'agent SNMP sur les nœuds

Pour assurer une réponse aux requêtes SNMP à partir du nœud réseau (switch ou routeur), il est nécessaire d'effectuer les configurations nécessaires sur ces équipements.

Il est à noter que la totalité des équipements sont du constructeur Cisco, ce qui facilitera la configuration soit les mêmes lignes de commande [wp4]. Il suffit donc d'accéder à distance sur l'équipement et ajouter la ligne de commande suivante

Snmp-server community ubcirouter RO

Snmp-server community ubciibm RO 10

Pour vérifier ces configurations, il suffit de saisir la commande show running-config sur le nœud réseau comme le montre la figure suivante :

Figure 23 Configuration SNMP sur un nœud réseau

2.2 Plateforme physique

Comme mentionné dans le chapitre conception, l'architecture adoptée est l'architecture 3tiers.Toutefois, les deux serveurs WEB est base de données seront hébergé ans le même serveur physique avec les caractéristiques matérielles suivantes :

- Système d'exploitation Microsoft Windows 7
- Mémoire 4 Go

3. Phase d'implémentation

3.1 Création de la base de données

Nous entamerons en premier lieu , la phase de création de la base donnée sous MySQL server qui stockera les données de supervision tel que état de lien (up/down) et charge de lien (bande passante) . On nommera cette base «DSI».

3.2 Création du script PHP

Le volet programmation dans notre travail se base sur des petit codes écrits en PHP , dont chacun a son rôle. La figure suivante montre une capture écran du script de connexion à la base de données.

```php
<?php
//---------------------------------------------------------------------

                    /*Entete  script*/
set_time_limit(1000);
ignore_user_abort(true);
session_start();
// Connexion à la base de données sur même serveur ( localhost) de nom "dsi"
    $BD_serveur     = "localhost";
    $BD_utilisateur = "root";
    $BD_motDePasse  = "";
    $BD_base        = "dsi";

    @mysql_pconnect($BD_serveur, $BD_utilisateur, $BD_motDePasse)
        or die("Impossible de se connecter au serveur de bases de données.");
    @mysql_select_db($BD_base)
        or die("Impossible de se connecter à la base de données.");
```

Figure 94 Script PHP de connexion à la base de données

3.3 Création des interfaces graphique

Cette étape concerne la préparation des différentes interfaces graphiques, et l'écriture des codes nécessaires permettant le lien entre ces interface afin d'avoir système compact et fonctionnel.

La figure suivante présente l'interface d'authentification

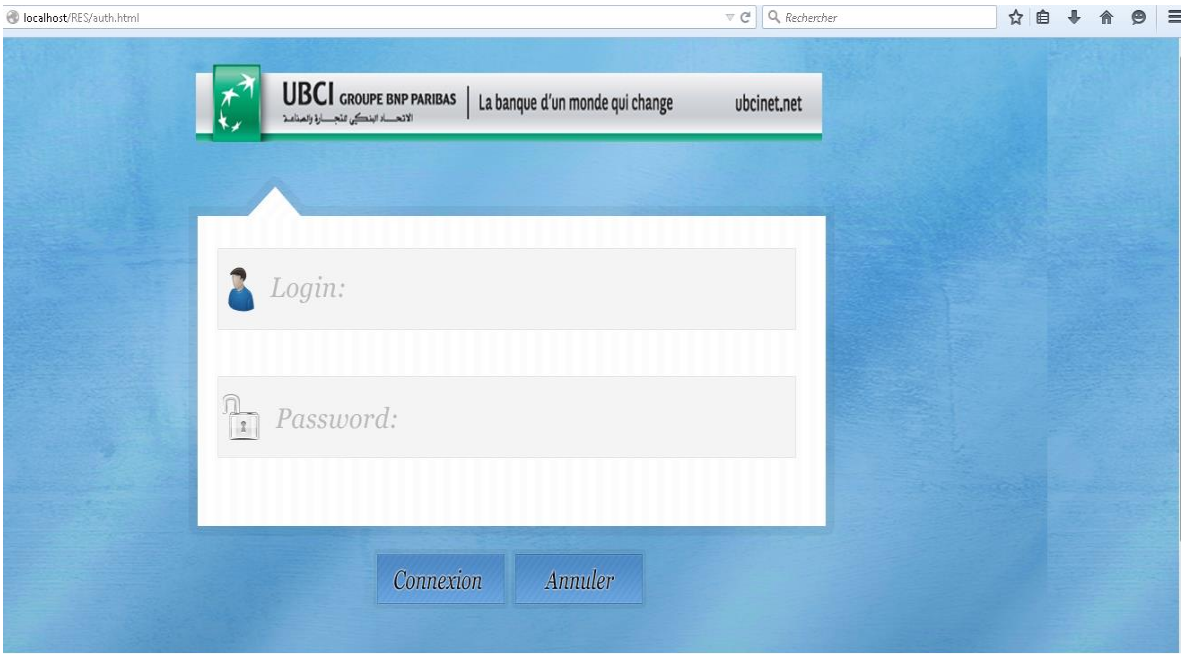


Figure 25 Interface graphique d'authentification

3.4 Résultat de test

Après avoir écrit les codes nécessaires et la manipulation des données et évènements SNMP dans la base donnée, il serait nécessaire d'effectuer les tests de bon fonctionnement de notre application

En effet, ces tests d'une grande importance car ils permettent d'évaluer le travail de conception qui est la tâche primordiale dans chaque développement. Ils permettent également de vérifier le bon fonctionnement de l'application.

Le script suivant permet d'afficher le résultat de test effectué sur une interface d'un routeur.

Figure 26 Script PHP de test du code

Au niveau de la base de données, les informations recueillies seront stockés dans les différentes tables.

IP	TrafficENTRANT	ETATINTERFACE	DATE1
10.142.31.2	2147483647	up(1)	24/06/2015 09:21:34
10.142.31.2	2147483647	up(1)	24/06/2015 09:21:48
10.142.31.2	2147483647	up(1)	24/06/2015 09:31:14
10.142.31.2	2147483647	up(1)	24/06/2015 09:33:38
10.142.31.2	2147483647	up(1)	24/06/2015 09:33:44
10.142.31.2	2147483647	up(1)	24/06/2015 09:34:10
10.142.31.2	2147483647	up(1)	24/06/2015 09:38:05
10.142.31.2	2147483647	up(1)	24/06/2015 09:38:53
10.142.31.2	2147483647	up(1)	24/06/2015 09:40:17
10.142.31.2	2147483647	up(1)	24/06/2015 09:50:18

Figure 27 Exemple de stockage dans la base de données

Conclusion

Dans ce chapitre nous avons énuméré les différentes étapes d'implémentation de notre application ainsi que quelques tests de vérification et de bon fonctionnement de l'application qui est le fruit de notre travail.

Conclusion Générale

Ce projet de fin d'étude a été réalisé au sein de l'Union Bancaire pour le Commerce et l'industrie dans le but de réaliser une application fiable de supervision de différents équipements réseau de la banque.

Ce travail a nécessité une étude approfondie de l'existant et les limites des outils utilisé par les administrateurs réseau et a justifié le choix de concevoir et développer notre application.

Une autre étude des différents aspects caractérisant l'opération de supervision et de management a été nécessaire pour entamer la phase de conception qui était la phase la plus importante.

La réalisation de cette application a nécessité l'utilisation de toute une plateforme de développement et la maitrise des notions avancées de la configuration et la gestion des équipements réseau et a donné à son achèvement de bons résultat et a été conforme aux attentes.

Néanmoins, cette application restera ouverture à des améliorations et des optimisation tel que l'ajout d'une composante logicielle permettant l'accès au cartographie réseau de la banque ainsi qu'au alertes à partir d'un client ***Android*** .

WEBOGRAPHIE

*[wp1] Critères de sélection d'une bonne infrastructure réseau d'après l'éditeur de solution de supervision Paessler . Mars 2015
https://assets.paessler.com/common/files/pdf/whitepaper/selection-criteria_fr.pdf
*[ub] Présentation de l'Union Bancaire pour le Commerce et l'Industrie UBCI , Avril 2015
http://ubci.tn/
*[wp2] Présentation de la plateforme de développement WampServer . Mai 2015
http://www.wampserver.com/
*[wp3] Présentation de l'ensemble de logiciel et de librairies Net-Snmp
https://fr.wikipedia.org/wiki/Net-SNMP
*Présentation du protocole de supervision standard SNMP
http://www.frameip.com/snmp/
http://ram-0000.developpez.com/tutoriels/reseau/SNMP/
http://www.linux-france.org/article/gvallee/snmp/snmp.html
*[wp4] Normes et notes configuration du protocole snmp sur les équipements de marque Cisco .
http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/configuration/guide/ffun_c/fcf014.html
https://www.manageengine.com/products/oputils/enable-snmp-cisco-router.html
http://www.technologuepro.com/reseaux/Configuration-Agent-SNMP/Configuration-SNMP-Routeur-CISCO.html
*Guide d'installation PHP
http://php.net/manual/en/install.php
liste des MIBs et de leurs objets est présente ici
http://www.simpleweb.org/ietf/mibs/
https://assets.nagios.com/presentations/nwcna2011/Mike%20Weber%20-%20NWC_SNMP.pdf

BIBLIOGHRAPIE

*Statut de l'UBCI , Document de présentation la Direction Système d'information de l'UBCI.

*Merise et UML pour la modélisation des systèmes d'information : un guide complet des études de cas. Joseph GABAY

*Computer Network : A Top-Down Approach . James F.KUROSE & Keith W.ROSS

*G. R. Ash, (1998), *Dynamic Routing in Telecommunications Networks*, McGraw Hill.

*Implementing Cisco IP Routing (ROUTE) : official course

*Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) : Official course

*Cisco OSPF Design Guide

*High Availability Campus Network Design – Routed Access Layer using EIGRP or OSPF : Cisco official documents

*HA Connectivity for servers and mainframes: NIC Teaming and OSA/OSPF Design

*BNP Paribas Global LAN Architecture Low Level Design

*Document d'Architecture Réseau LAN & WAN UBCI – Senda Boukef (IBM Tunisie)

*Cisco Doucments « Layer 3 MPLS VPN Enterprise Consumer Guide Version 2 » - Réf. : OL-8851-01

Annexe 1

Document de normalisation SNMP par l'EITF

Network Working Group
Request for Comments: 1157
Obsoletes: RFC 1098

J. Case
SNMP Research
M. Fedor
Performance Systems International
M. Schoffstall
Performance Systems International
J. Davin
MIT Laboratory for Computer Science
May 1990

A Simple Network Management Protocol (SNMP)

Table of Contents

1. Status of this Memo

This RFC is a re-release of RFC 1098, with a changed "Status of this Memo" section plus a few minor typographical corrections. This memo defines a simple protocol by which management information for a network element may be inspected or altered by logically remote users. In particular, together with its companion memos which describe the structure of management information along with the management information base, these documents provide a simple, workable architecture and system for managing TCP/IP-based internets and in particular the Internet.

The Internet Activities Board recommends that all IP and TCP implementations be network manageable. This implies implementation of the Internet MIB (RFC-1156) and at least one of the two recommended management protocols SNMP (RFC-1157) or CMOT (RFC-1095). It should be noted that, at this time, SNMP is a full Internet standard and CMOT is a draft standard. See also the Host and Gateway Requirements RFCs for more specific information on the applicability of this standard.

Please refer to the latest edition of the "IAB Official Protocol Standards" RFC for current information on the state and status of standard Internet protocols.

Distribution of this memo is unlimited.

2. Introduction

As reported in RFC 1052, IAB Recommendations for the Development of Internet Network Management Standards [1], a two-prong strategy for network management of TCP/IP-based internets was undertaken. In the short-term, the Simple Network Management Protocol (SNMP) was to be used to manage nodes in the Internet community. In the long-term, the use of the OSI network management framework was to be examined. Two documents were produced to define the management information: RFC 1065, which defined the Structure of Management Information (SMI) [2], and RFC 1066, which defined the Management Information Base (MIB) [3]. Both of these documents were designed so as to be

compatible with both the SNMP and the OSI network management

framework.This strategy was quite successful in the short-term:

Internet-based

network management technology was fielded, by both the research and commercial communities, within a few months. As a result of this, portions of the Internet community became network manageable in a

timely fashion.

As reported in RFC 1109, Report of the Second Ad Hoc Network Management Review Group [4], the requirements of the SNMP and the OSI

network management frameworks were more different than anticipated. As such, the requirement for compatibility between the SMI/MIB and both frameworks was suspended. This action permitted the operational network management framework, the SNMP, to respond to new operational needs in the Internet community by producing documents defining new MIB items.

The IAB has designated the SNMP, SMI, and the initial Internet MIB to be full "Standard Protocols" with "Recommended" status. By this action, the IAB recommends that all IP and TCP implementations be network manageable and that the implementations that are network manageable are expected to adopt and implement the SMI, MIB, and SNMP.

As such, the current network management framework for TCP/IP- based internets consists of: Structure and Identification of Management Information for TCP/IP-based Internets, which describes how managed objects contained in the MIB are defined as set forth in RFC 1155 [5]; Management Information Base for Network Management of TCP/IP-based Internets, which describes the managed objects contained in the MIB as set forth in RFC 1156 [6]; and, the Simple Network Management Protocol, which defines the protocol used to manage these objects, as set forth in this memo.

As reported in RFC 1052, IAB Recommendations for the Development of Internet Network Management Standards [1], the Internet Activities Board has directed the Internet Engineering Task Force (IETF) to create two new working groups in the area of network management. One group was charged with the further specification and definition of elements to be included in the Management Information Base (MIB). The other was charged with defining the modifications to the Simple Network Management Protocol (SNMP) to accommodate the short-term needs of the network vendor and operations communities, and to align with the output of the MIB working group.

The MIB working group produced two memos, one which defines a Structure for Management Information (SMI) [2] for use by the managed

objects contained in the MIB. A second memo [3] defines the list of managed objects.

The output of the SNMP Extensions working group is this memo, which incorporates changes to the initial SNMP definition [7] required to attain alignment with the output of the MIB working group. The changes should be minimal in order to be consistent with the IAB's directive that the working groups be "extremely sensitive to the need to keep the SNMP simple

" Although considerable care and debate hasgone into the changes to the SNMP which are reflected in this memo,

the resulting protocol is not backwardly-compatible with its

predecessor, the Simple Gateway Monitoring Protocol (SGMP) [8]. Although the syntax of the protocol has been altered, the original philosophy, design decisions, and architecture remain intact. In order to avoid confusion, new UDP ports have been allocated for use

by the protocol described in this memo.

3. The SNMP Architecture

Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements.

3.1. Goals of the Architecture

The SNMP explicitly minimizes the number and complexity of management functions realized by the management agent itself. This goal is attractive in at least four respects:

(1) The development cost for management agent software necessary to support the protocol is accordingly reduced.

(2) The degree of management function that is remotely supported is accordingly increased, thereby admitting fullest use of internet resources in the management task.

(3) The degree of management function that is remotely supported is accordingly increased, thereby imposing the fewest possible restrictions on the form and sophistication of management tools.

(4) Simplified sets of management functions are easily understood and used by developers of network management tools.

A second goal of the protocol is that the functional paradigm for monitoring and control be sufficiently extensible to accommodate additional, possibly unanticipated aspects of network operation and management.

A third goal is that the architecture be, as much as possible, independent of the architecture and mechanisms of particular hosts or particular gateways.

3.2. Elements of the Architecture

The SNMP architecture articulates a solution to the network management problem in terms of:

(1) the scope of the management information communicated by the protocol,

(2) the representation of the management information communicated by the protocol,

(3) operations on management information supported by the protocol,

(4) the form and meaning of exchanges among management entities,

(5) the definition of administrative relationships among management entities, and

(6) the form and meaning of references to management information.

3.2.1. Scope of Management Information

The scope of the management information communicated by operation of the SNMP is exactly that represented by instances of all non-aggregate object types either defined in Internet-standard MIB or defined elsewhere according to the conventions set forth in Internet-standard SMI [5].

Support for aggregate object types in the MIB is neither required for conformance with the SMI nor realized by the SNMP.

3.2.2. Representation of Management Information

Management information communicated by operation of the SNMP is represented according to the subset of the ASN.1 language [9] that is specified for the definition of non-aggregate types in the SMI.

The SGMP adopted the convention of using a well-defined subset of the ASN.1 language [9]. The SNMP continues and extends this tradition by utilizing a moderately more complex subset of ASN.1 for describing managed objects and for describing the protocol data units used for managing those objects. In addition, the desire to ease eventual transition to OSI-based network management protocols led to the definition in the ASN.1 language of an Internet-standard Structure of Management Information (SMI) [5] and Management Information Base (MIB) [6]. The use of the ASN.1 language, was, in part, encouraged by the successful use of ASN.1 in earlier efforts, in particular, the SGMP. The restrictions on the use of ASN.1 that are part of the SMI contribute to the simplicity espoused and validated by experience with the SGMP.
Also for the sake of simplicity, the SNMP uses only a subset of the basic encoding rules of ASN.1 [10]. Namely, all encodings use the definite-length form. Further, whenever permissible, non-constructor encodings are used rather than constructor encodings. This restriction applies to all aspects of ASN.1 encoding, both for the top-level protocol data units and the data objects they contain.

3.2.3. Operations Supported on Management Information

The SNMP models all management agent functions as alterations or

inspections of variables. Thus, a protocol entity on a logically remote host (possibly the network element itself) interacts with the

management agent resident on the network element in order to retrieve (get) or alter (set) variables. This strategy has at least two positive consequences:

(1) It has the effect of limiting the number of essential management functions realized by the management agent to two: one operation to assign a value to a specified configuration or other parameter and another to retrieve such a value.

(2) A second effect of this decision is to avoid introducing into the protocol definition support for imperative management commands: the number of such commands is in practice ever-increasing, and the semantics of such commands are in general arbitrarily complex.

The strategy implicit in the SNMP is that the monitoring of network state at any significant level of detail is accomplished primarily by polling for appropriate information on the part of the monitoring center(s). A limited number of unsolicited messages (traps) guide the timing and focus of the polling. Limiting the number of unsolicited messages is consistent with the goal of simplicity and minimizing the amount of traffic generated by the network management function.

The exclusion of imperative commands from the set of explicitly supported management functions is unlikely to preclude any desirable management agent operation. Currently, most commands are requests either to set the value of some parameter or to retrieve such a value, and the function of the few imperative commands currently supported is easily accommodated in an asynchronous mode by this management model. In this scheme, an imperative command might be realized as the setting of a parameter value that subsequently triggers the desired action. For example, rather than implementing a "reboot command," this action might be invoked by simply setting a parameter indicating the number of seconds until system reboot.

3.2.4. Form and Meaning of Protocol Exchanges

The communication of management information among management entities is realized in the SNMP through the exchange of protocol messages. The form and meaning of those messages is defined below in Section 4.

Consistent with the goal of minimizing complexity of the management agent, the exchange of SNMP messages requires only an unreliable datagram service, and every message is entirely and independently represented by a single transport datagram. While this document specifies the exchange of messages via the UDP protocol [11], the mechanisms of the SNMP are generally suitable for use with a wide variety of transport services.

3.2.5. Definition of Administrative Relationships

The SNMP architecture admits a variety of administrative relationships among entities that participate in the protocol. The entities residing at management stations and network elements which communicate with one another using the SNMP are termed SNMP application entities. The peer processes which implement the SNMP,

and thus support the SNMP application entities, are termed protocol entities.

A pairing of an SNMP agent with some arbitrary set of SNMP application entities is called an SNMP community. Each SNMP community is named by a string of octets, that is called the community name for said community.

An SNMP message originated by an SNMP application entity that in fact belongs to the SNMP community named by the community component of said message is called an authentic SNMP message. The set of rules by which an SNMP message is identified as an authentic SNMP message for a particular SNMP community is called an authentication scheme. An implementation of a function that identifies authentic SNMP messages according to one or more authentication schemes is called an authentication service.

Clearly, effective management of administrative relationships among SNMP application entities requires authentication services that (by the use of encryption or other techniques) are able to identify authentic SNMP messages with a high degree of certainty. Some SNMP implementations may wish to support only a trivial authentication service that identifies all SNMP messages as authentic SNMP messages.

For any network element, a subset of objects in the MIB that pertain to that element is called a SNMP MIB view. Note that the names of the object types represented in a SNMP MIB view need not belong to a single sub-tree of the object type name space.

An element of the set { READ-ONLY, READ-WRITE } is called an SNMP access mode.

A pairing of a SNMP access mode with a SNMP MIB view is called an SNMP community profile. A SNMP community profile represents specified access privileges to variables in a specified MIB view. For every variable in the MIB view in a given SNMP community profile, access to that variable is represented by the profile according to the following conventions:

(1) if said variable is defined in the MIB with "Access:" of "none," it is unavailable as an operand for any operator;

(2) if said variable is defined in the MIB with "Access:" of "read-write" or "write-only" and the access mode of the given profile is READ-WRITE, that variable is available as an operand for the get, set, and trap operations;

(3) otherwise, the variable is available as an operand for the get and trap operations.

(4) In those cases where a "write-only" variable is an operand used for the get or trap operations, the value given for the variable is implementation-specific.

A pairing of a SNMP community with a SNMP community profile is called a SNMP access policy. An access policy represents a specified community profile afforded by the SNMP agent of a specified SNMP community to other members of that community. All administrative

relationships among SNMP application entities are architecturally defined in terms of SNMP access policies.

For every SNMP access policy, if the network element on which the SNMP agent for the specified SNMP community resides is not that to which the MIB view for the specified profile pertains, then that policy is called a SNMP proxy access policy. The SNMP agent associated with a proxy access policy is called a SNMP proxy agent.

While careless definition of proxy access policies can result in management loops, prudent definition of proxy policies is useful in at least two ways:

(1) It permits the monitoring and control of network elements which are otherwise not addressable using the management protocol and the transport protocol. That is, a proxy agent may provide a protocol conversion function allowing a management station to apply a consistent management

framework to all network elements, including devices such as modems, multiplexors, and other devices which support different management frameworks.

(2) It potentially shields network elements from elaborate access control policies. For example, a proxy agent may implement sophisticated access control whereby diverse subsets of variables within the MIB are made accessible to different management stations without increasing the complexity of the network element.

By way of example, Figure 1 illustrates the relationship between management stations, proxy agents, and management agents. In this example, the proxy agent is envisioned to be a normal Internet Network Operations Center (INOC) of some administrative domain which has a standard managerial relationship with a set of management agents.

Domain: the administrative domain of the element
PCommunity: the name of a community utilizing a proxy agent
DCommunity: the name of a direct community

Figure 1
Example Network Management Configuration

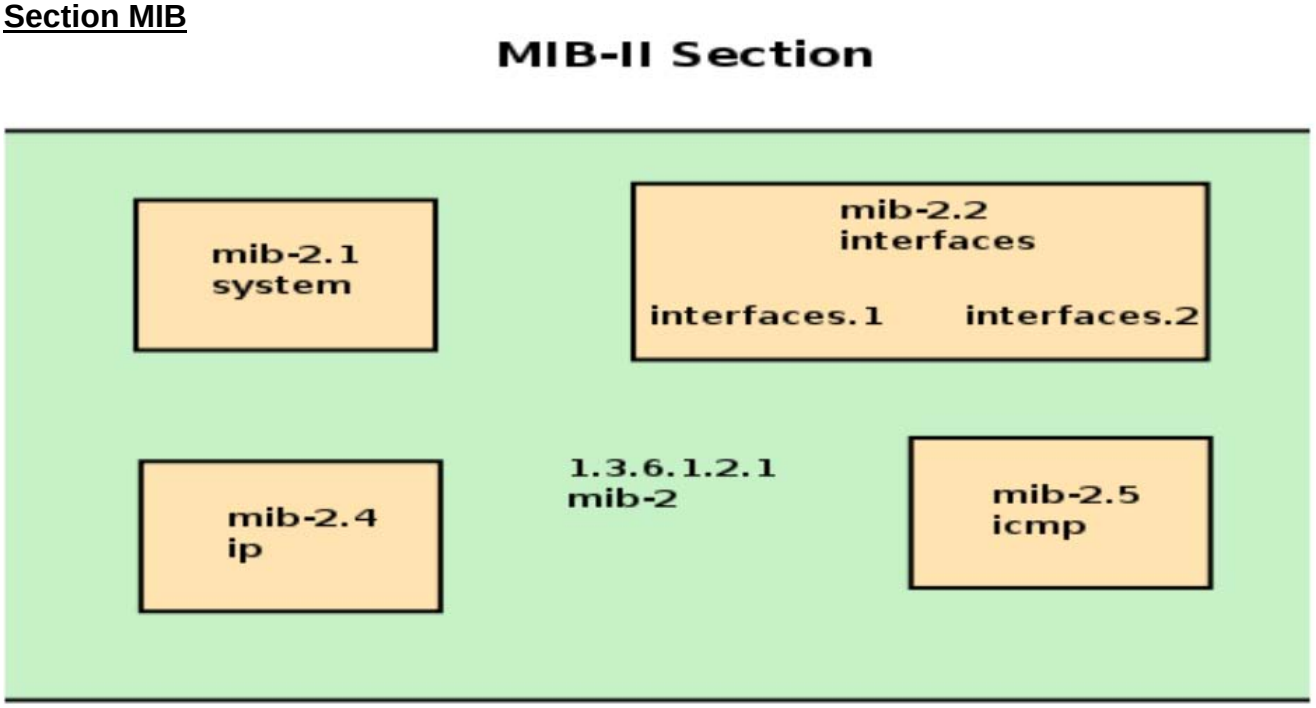
Annexe 2:

Libraires SNMP

SNMP Library Numbering: Arbre hierarchique

1 – iso
1.3 – org
1.3.6 – dod
1.3.6.1 – internet
1.3.6.1.1 – directory
1.3.6.1.2 – mgmt
1.3.6.1.2.1 mib-2
1.3.6.1.3 – experimental
1.3.6.1.4 – private
1.3.6.1.5 – security
1.3.6.1.6 - SNMPv

Section MIB



Management Information Base (MIB)

MIBS provide a list of available OIDS, that is why in library sense it is a section.

```
IfEntry ::=
    SEQUENCE {
        ifIndex                 InterfaceIndex,
        ifDescr                 DisplayString,
        ifType                  IANAifType,
```

```
ifMtu                   Integer32,
ifSpeed                 Gauge32,
ifPhysAddress           PhysAddress,
ifAdminStatus           INTEGER,
ifOperStatus            INTEGER,
ifLastChange            TimeTicks,
ifInOctets              Counter32,
ifInUcastPkts           Counter32,
ifInNUcastPkts          Counter32,   deprecated
ifInDiscards            Counter32,
ifInErrors              Counter32,
ifInUnknownProtos       Counter32,
ifOutOctets             Counter32,
ifOutUcastPkts          Counter32,
ifOutNUcastPkts         Counter32,   deprecated
ifOutDiscards           Counter32,
ifOutErrors             Counter32,
ifOutQLen               Gauge32,     deprecated
ifSpecific              OBJECT IDENTIFIER  deprecated
```

Annexe 3:

MIB IF-INTERFACE

Statistics for **MIB IF-MIB:**

Objects: **100**	OIDS: **91**	Object Groups: **13**
Traps: **0**	Notifications: **2**	Notification Groups: **1**
Tables: **5**	Tabulars: **53**	Scalars/Other: **17**

Object Name	Object Identifier
interfaces	1.3.6.1.2.1.2
ifNumber	1.3.6.1.2.1.2.1
ifTable	1.3.6.1.2.1.2.2
ifEntry	1.3.6.1.2.1.2.2.1
ifIndex	1.3.6.1.2.1.2.2.1.1
ifInOctets	1.3.6.1.2.1.2.2.1.10
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12
ifInDiscards	1.3.6.1.2.1.2.2.1.13
ifInErrors	1.3.6.1.2.1.2.2.1.14
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15
ifOutOctets	1.3.6.1.2.1.2.2.1.16
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18
ifOutDiscards	1.3.6.1.2.1.2.2.1.19
ifDescr	1.3.6.1.2.1.2.2.1.2
ifOutErrors	1.3.6.1.2.1.2.2.1.20
ifOutQLen	1.3.6.1.2.1.2.2.1.21
ifSpecific	1.3.6.1.2.1.2.2.1.22
ifType	1.3.6.1.2.1.2.2.1.3
ifMtu	1.3.6.1.2.1.2.2.1.4
ifSpeed	1.3.6.1.2.1.2.2.1.5
ifPhysAddress	1.3.6.1.2.1.2.2.1.6
ifAdminStatus	1.3.6.1.2.1.2.2.1.7

ifOperStatus	1.3.6.1.2.1.2.2.1.8
ifLastChange	1.3.6.1.2.1.2.2.1.9
ifMIB	1.3.6.1.2.1.31
ifMIBObjects	1.3.6.1.2.1.31.1
ifXTable	1.3.6.1.2.1.31.1.1
ifXEntry	1.3.6.1.2.1.31.1.1.1
ifName	1.3.6.1.2.1.31.1.1.1.1
ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10
ifHCOutUcastPkts	1.3.6.1.2.1.31.1.1.1.11
ifHCOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.12
ifHCOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.13
ifLinkUpDownTrapEnable	1.3.6.1.2.1.31.1.1.1.14
ifHighSpeed	1.3.6.1.2.1.31.1.1.1.15
ifPromiscuousMode	1.3.6.1.2.1.31.1.1.1.16
ifConnectorPresent	1.3.6.1.2.1.31.1.1.1.17
ifAlias	1.3.6.1.2.1.31.1.1.1.18
ifCounterDiscontinuityTime	1.3.6.1.2.1.31.1.1.1.19
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6
ifHCInUcastPkts	1.3.6.1.2.1.31.1.1.1.7
ifHCInMulticastPkts	1.3.6.1.2.1.31.1.1.1.8
ifHCInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.9
ifStackTable	1.3.6.1.2.1.31.1.2
ifStackEntry	1.3.6.1.2.1.31.1.2.1
ifStackHigherLayer	1.3.6.1.2.1.31.1.2.1.1
ifStackLowerLayer	1.3.6.1.2.1.31.1.2.1.2
ifStackStatus	1.3.6.1.2.1.31.1.2.1.3
ifTestTable	1.3.6.1.2.1.31.1.3
ifTestEntry	1.3.6.1.2.1.31.1.3.1
ifTestId	1.3.6.1.2.1.31.1.3.1.1
ifTestStatus	1.3.6.1.2.1.31.1.3.1.2
ifTestType	1.3.6.1.2.1.31.1.3.1.3
ifTestResult	1.3.6.1.2.1.31.1.3.1.4
ifTestCode	1.3.6.1.2.1.31.1.3.1.5
ifTestOwner	1.3.6.1.2.1.31.1.3.1.6
ifRcvAddressTable	1.3.6.1.2.1.31.1.4

ifRcvAddressEntry	1.3.6.1.2.1.31.1.4.1
ifRcvAddressAddress	1.3.6.1.2.1.31.1.4.1.1
ifRcvAddressStatus	1.3.6.1.2.1.31.1.4.1.2
ifRcvAddressType	1.3.6.1.2.1.31.1.4.1.3
ifTableLastChange	1.3.6.1.2.1.31.1.5
ifStackLastChange	1.3.6.1.2.1.31.1.6
ifConformance	1.3.6.1.2.1.31.2
ifGroups	1.3.6.1.2.1.31.2.1
ifGeneralGroup	1.3.6.1.2.1.31.2.1.1
ifGeneralInformationGroup	1.3.6.1.2.1.31.2.1.10
ifStackGroup2	1.3.6.1.2.1.31.2.1.11
ifOldObjectsGroup	1.3.6.1.2.1.31.2.1.12
ifCounterDiscontinuityGroup	1.3.6.1.2.1.31.2.1.13
linkUpDownNotificationsGroup	1.3.6.1.2.1.31.2.1.14
ifFixedLengthGroup	1.3.6.1.2.1.31.2.1.2
ifHCFixedLengthGroup	1.3.6.1.2.1.31.2.1.3
ifPacketGroup	1.3.6.1.2.1.31.2.1.4
ifHCPacketGroup	1.3.6.1.2.1.31.2.1.5
ifVHCPacketGroup	1.3.6.1.2.1.31.2.1.6
ifRcvAddressGroup	1.3.6.1.2.1.31.2.1.7
ifTestGroup	1.3.6.1.2.1.31.2.1.8
ifStackGroup	1.3.6.1.2.1.31.2.1.9
ifCompliances	1.3.6.1.2.1.31.2.2
ifCompliance	1.3.6.1.2.1.31.2.2.1
ifCompliance2	1.3.6.1.2.1.31.2.2.2
ifCompliance3	1.3.6.1.2.1.31.2.2.3
linkDown	1.3.6.1.6.3.1.1.5.3
linkUp	1.3.6.1.6.3.1.1.5.4

Résumé

Ce projet a été réalisée au sein du département des Systèmes d'Information de l'Union Bancaire pour le commerce est l'industrie à Tunis. Il s'agit d'un projet de fin d'études pour l'obtention du Master professionnel en Nouvelles Technologies des Télécommunications et Réseaux à l'Université Virtuelle de Tunis qui consiste à réaliser une application web pour la supervision du réseau de l'UBCI, basée sur le protocole SNMP.

Cette application implémente plusieurs fonctionnalités qui permettent à l'administrateur réseau de gérer son réseau d'une manière fiable te simple tout en gardant le réseau fonctionnel.

Mots Clés:

Supervision, disponibilité, Architecture 3-tiers, SNMP, Trap, Base de Données, PHP

Abstract

This report was prepared as part of the final project study for obtaining the Master Degree in New Information Technologies. The aim was to develop a web application for monitoring the whole Network of UBCI based on SNMP.

This application implements several features that enable the network administrator to manage its network in a reliable and simple while keeping the network functional.

Keywords :

Supervision, Availability, 3-tier Architecture, SNMP , Trap, Database , PHP