

**Version <1.0>**

Date de création : *15/03/2010*

Dernière MAJ : *27/03/2010*

Etat : *validé*

**Auteurs**

Arthur Gervais   
Thomas Guillory   
Jonathan Giroux  
Dragos Constantin  
Mathieu Andre  
Pavel Efros

Projet d’Administration Réseau

Sommaire

[I. Introduction 3](#_Toc257480656)

[II. Objectifs et motivations pour la nouvelle architecture 3](#_Toc257480657)

[A. Au niveau technologiques et utilisation de L’architecture 3](#_Toc257480658)

[B. Au niveau maintenance et configuration 3](#_Toc257480659)

[III. Organisation de l’architecture 5](#_Toc257480660)

[A. Principe 5](#_Toc257480661)

[1. Schéma global 5](#_Toc257480662)

[2. Organisation de la solution dans la version multi-sites 5](#_Toc257480663)

[3. Element clefs du cahier des charges et impact sur la solution 7](#_Toc257480664)

[B. Description de la solution 9](#_Toc257480665)

[IV. Solution pour le nommage et l’association des Adresses IP 12](#_Toc257480666)

[A. Gestion de la politique de nommage 12](#_Toc257480667)

[1. Definition de la politique 12](#_Toc257480668)

[2. Organisation du service de DNS multi-domaines 13](#_Toc257480669)

[3. Mise en place de l’infrastructure de DHCP 14](#_Toc257480670)

[4. Procédure d’ajout de matériel 16](#_Toc257480671)

[V. Mise en place d’un système d’administration 16](#_Toc257480672)

[A. Principe de la solution proposée 16](#_Toc257480673)

[B. Comparaison des différentes solutions logicielles 18](#_Toc257480674)

[1. Identification de critères de comparaison 18](#_Toc257480675)

[2. Tableau récapitulatif 18](#_Toc257480676)

[C. Organisation de la solution 19](#_Toc257480677)

[1. Besoins en hardware et software 19](#_Toc257480678)

[2. Gestion des configurations 19](#_Toc257480679)

[3. Remarques 21](#_Toc257480680)

[VI. Annexe 1 – TEST 21](#_Toc257480681)

[A. Paramètrage générale 21](#_Toc257480682)

[B. Résolution Servif-baie 22](#_Toc257480683)

[C. Résolution Mail 23](#_Toc257480684)

[D. Résolution d’un IP 23](#_Toc257480685)

[E. Résolution de www.google.fr 24](#_Toc257480686)

[VII. Annexe 2 – Monitoring reseau 24](#_Toc257480687)

[A. Utilisation des commandes de monitoring 25](#_Toc257480688)

[1. Détermination des paramètres réseaux 25](#_Toc257480689)

[*2.* Commande *netstat* 26](#_Toc257480690)

[B. Sniffer de paquet et ping sur IFSWITCH06 39](#_Toc257480691)

[C. Serveur nagios et MiB 42](#_Toc257480692)

[D. Topologie par *traceroute* 43](#_Toc257480693)

[VIII. Annexe 3 – NAGIOS 45](#_Toc257480694)

[A. Introduction 45](#_Toc257480695)

[B. Remarques sur le tutoriel d’installation 45](#_Toc257480696)

[C. Fichiers de configuration 45](#_Toc257480697)

[1. NSC.ini 45](#_Toc257480698)

[2. Nagios.cfg 46](#_Toc257480699)

[3. windows.cfg 46](#_Toc257480700)

[4. windowsIF.cfg 47](#_Toc257480701)

[D. Captures d’ecran 49](#_Toc257480702)

[E. Analyse critique 51](#_Toc257480703)

[1. Sur l’installation 51](#_Toc257480704)

[2. Sur l’utilisation 51](#_Toc257480705)

[IX. Annexe 4 – Interpretation des Courbes de MRTG 52](#_Toc257480706)

[A. « Weekly » Graph (30 minute average) 52](#_Toc257480707)

[B. « Daily » Graph (5 minute average) 52](#_Toc257480708)

[C. Moyenne pourcentage de charge CPU 52](#_Toc257480709)

[D. « Daily Graph » (5 Minute average) 52](#_Toc257480710)

[E. Graphe annuel (moyenne sur 1 jour) 52](#_Toc257480711)

[F. Gestion du Spam : « weekly » graph (moyenne sur 30 minutes) 53](#_Toc257480712)

# Introduction

Suite à l’évolution de l’architecture du réseau informatique de l’AIP RAO et de son administration, notre équipe a été sollicitée afin de proposer une solution qui réponde aux nouveaux besoins induits par cette transformation. Ce présent dossier vise à présenter l’ensemble des aspects de cette solution originale du plus général (architecture matérielle) au plus précis (politique de nommage). Les annexes fourniront également des conseils et des exemples d’utilisation pour faciliter la mise en place de la solution.

# Objectifs et motivations pour la nouvelle architecture

La direction de l’AIPRAO a retenu les objectifs suivants pour l’infrastructure :

## Au niveau technologiques et utilisation de L’architecture

* L’architecture doit favoriser les regroupements de moyens.
  + Les différents sites, à savoir l’AIP, GE et Roanne, doivent pouvoir être interconnectés. Une plateforme industrielle virtuelle doit donc pouvoir être répartie sur plusieurs plateformes physiquement éloignées. Ceci permettra de minimiser les transports pour les usagers.
* L’architecture doit permettre d’élargir au maximum les possibilités de travail à distance.
  + L’infrastructure doit permettre de surveiller visuellement le système et d’avoir un contrôle à distance.
* Sécurité de l’architecture
  + Les mécanismes d’accès à distance à l’infrastructure ne doivent en aucun cas mettre en péril la sécurité du système entier. Il faut donc veiller à mettre en place un système de sécurisation d’accès.
* Réduction des pertes de performance
  + La nouvelle architecture doit permettre de réduire les pertes de performance actuelles du réseau.
* Assurance du fonctionnement des TP
  + L’architecture proposée doit permettre de garantir au mieux la bonne disponibilité des platines industrielles pour un usage en TP.
* Mobilité des plateformes
  + Il doit être possible de déplacer un équipement d’une plateforme vers un autre site.

## Au niveau maintenance et configuration

* L’architecture doit être prête à pouvoir évoluer. Un ajout d’un nouveau serveur, ou d’un équipement industriel ne doit pas poser de problèmes insurmontables pour les administrateurs réseau.
* L’architecture doit permettre de superviser le réseau global. Pour ceci un système d’administration doit être mis en place.

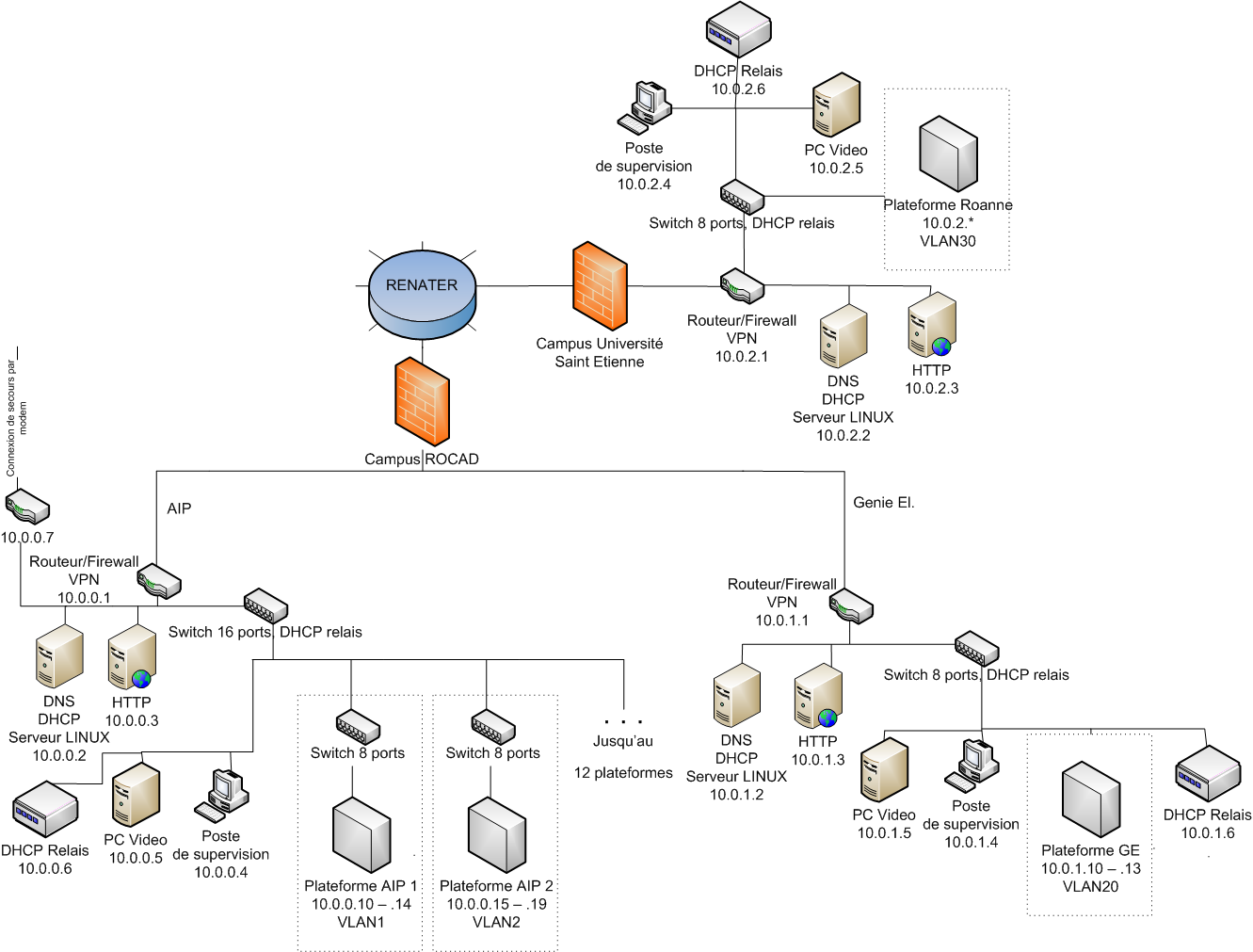
# Organisation de l’architecture

Dans la suite nous vous présentons notre solution pour la version multi-sites. Vous allez voir pourquoi notre solution est simple, évolutive tout en restant sécurisée.

## Principe

### Schéma global

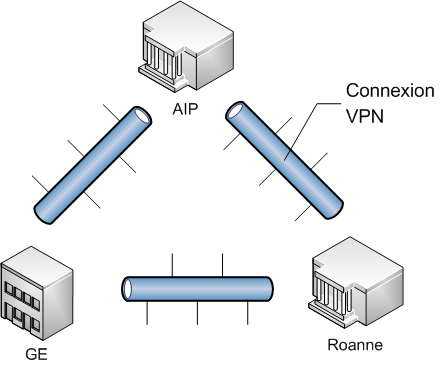
Afin que vous puissiez le mieux comprendre la description de l’architecture voici le schéma global que nous proposons avec les 3 sites respectivement AIP, GE et Roanne.



### Organisation de la solution dans la version multi-sites

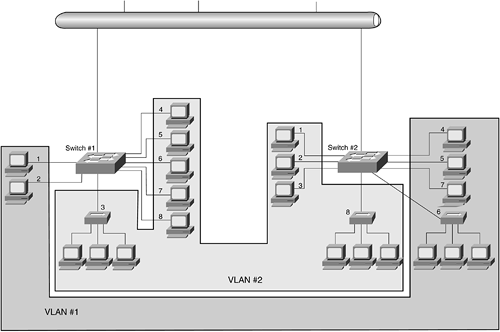
Notre architecture est conçue pour fonctionner dans une version multi-sites et également dans une version mono-site si jamais la liaison avec les sites distants est rompue.

Les différents sites seront reliés par un VPN sécurisant l’accès et permettant d’authentifier au mieux les différents utilisateurs accédant aux plateformes industrielles. Ce VPN s’établit donc comme suit entre les sites :



Les différents sites peuvent s’interconnecter grâce au routeur S@N que nous employons. Ainsi chaque site aura au moins deux connexions VPN en parallèle. Le but est que tout site se trouve dans un VPN global aux sites afin de partager les ressources.

Un étudiant ou professeur pourra à tout moment se connecter depuis l’extérieur vers un des sites et ainsi avoir accès aux ressources qu’il héberge.

L’architecture d’un site donné est découpée en plusieurs VLAN. Chaque plateforme industrielle sera attribuée à un propre VLAN afin de bien séparer l’architecture et d’augmenter la sécurité ainsi que la performance du réseau entier. Il faut garder en tête qu’un VLAN n’est pas plus qu’un sous réseau virtuel. Le principal avantage de ce type de sous-réseau est que les différents composants qui lui appartiennent ne doivent pas forcément être situés sur le même routeur ou switch de layer 3. Ceci permettra donc par la suite de créer des VLAN qui regroupent dans un même sous-réseau des équipements qui pourront se trouver sur des sites distants.

L’association des adresses IP se fera par le biais d’un serveur DHCP se trouvant sur chaque site. Les serveurs DHCP auront toute une configuration identique et on utilisera un DHCP relais afin d’associer les adresses IP. Les DHCP relais peuvent examiner de quels sous-réseaux les demandes DHCP viennent et associer en fonction les adresses IP.

Ainsi un adressage inter-site est possible en gardant les équipements dans un VPN global. Les détails de l’adressage seront expliqués dans la partie « Mise en place de l’infrastructure de DHCP ».

En ce qui concerne le DNS nous avons choisie d’installer un serveur DNS par site. Comme pour la configuration DHCP les configurations DNS seront donc partager sur les différentes sites. Pour connaitre la politique de nommage je vous prie de vous référer au paragraphe « Gestion de la politique de nommage ».

La configuration des serveurs DHCP et DNS sera mise à jour et partagée entre les sites par le biais d’un système de monitoring comme Nagios par exemple.

### Element clefs du cahier des charges et impact sur la solution

En reprenant les éléments clefs du cahier des charges, nous allons décrire par la suite pourquoi notre solution y est parfaitement adaptée.

#### L’architecture doit favoriser les regroupements de moyens

Notre architecture permet de regrouper des équipements industriels qui se trouvent physiquement très éloignés ou sur un site distant.

Ceci est possible grâce-à notre architecture en VLAN. Imaginons que nous avons un site A avec une plateforme industrielle comportant 5 équipements. Sur un site B éloigné nous avons un équipement que nous aimerons utiliser par le biais du réseau avec la plateforme du site A.

Notre solution est donc d’interconnecter ces équipements en les mettant dans un VLAN qui les regroupe. Ainsi ils se trouveront « logiquement » dans le même sous-réseau et le regroupement de moyens est réalisé. La configuration des VLAN se fera au niveau du switch principal de chaque site. Ceci est une opération qui peut se faire facilement via l’interface web du switch et nécessite aucun déplacement ou branchement matériel.

#### L’architecture doit permettre d’Élargir au maximum les possibilités de travail À distance

Grâce à un serveur HTTP sur chaque site, il est possible de se connecter avec n’importe quel navigateur internet sur les différents sites. Le serveur HTTP offrira un accès aux platines industrielles via une interface web. La supervision des manipulations se fera en plus par le biais de caméras vidéo.

Le serveur HTTP sera accessible depuis n’importe quel poste qui est connecté à Internet. La seule condition pour pouvoir se connecter au serveur sera d’être authentifiée auparavant par le VPN. Cette manière de se connecter garantit que notre architecture ne possède qu’un seul point d’entrée (la connexion VPN) et assure ainsi la sécurisation des points d’accès ce qui est un des points importants du cahier des charges. Ceci nous ramène donc au point suivant.

#### Sécurité de l’architecture

Les mécanisme d’accès à distance permettent de contrôler pleinement l’architecture industrielle, mais ne doivent en aucun cas mettre en péril la sécurité. Nous avons donc choisi de n’avoir qu’un seul point d’entrée vers les sites à travers le VPN. Le VPN lui-même sera configuré afin de garantir le plus de sécurité possible. Pour en savoir plus, veuillez vous référer au paragraphe « Description de la solution ».

Les VLAN permettent également d’augmenter la sécurité du réseau entier grâce à une configuration plus facile et un découpage logique.

#### Réduction des pertes de performance

L’ancien système a souffert de nombreuses pertes de performance dû au non-découpage en sous-réseaux. Notre solution emploie un VLAN par plateforme industrielle ce qui limite la portée des paquets broadcast sur le VLAN. Ainsi le réseau entier bénéficiera d’une augmentation des performances globales.

#### Mobilité des platformes

Grâce à notre politique de nommage (voir paragraphe «Gestion de la politique de nommage ») et l’attribution des adresses IP (voir paragraphe « Mise en place de l’infrastructure de DHCP ») tout équipement ou plateforme industrielle peut changer d’emplacement et fonctionner instantanément après avoir été connecté au sous réseau du site respectif. En effet, notre système est capable de reconnaître quand une machine a changé d’endroit.

Ceci garantit donc la mobilité des plateformes comme exigé par le cahier de charges.

#### Assurance du fonctionnement des TPs

Un système d’administration est intégré dans notre architecture afin de pouvoir superviser au mieux l’état et le fonctionnement actuel. Ainsi une maintenance préventive est installé ce qui nous permet de garantir au mieux une bonne accessibilité des composants industriel pour un usage en TP.

Veuillez vous référer au paragraphe « Mise en place d’un système d’administration ».

En cas de pannes du réseau il est possible d’utiliser une connexion de secours avec le router FSX RAS 2000 via le réseau modem. Ceci permettra de debugger au plus vite le réseau et augmente ainsi la disponibilité.

#### Évolution de l’Architecture

Notre architecture est simple et reste pourtant prête à évoluer. L’avantage des VLAN est qu’il est facile de rajouter de nouveaux composants et de les réorganiser dans un sous-réseau sans pourtant modifier des composants matériels.

Sachant qu’on aurait aussi pu se baser sur des adresses routables uniquement sur le réseau RENATER il semble pourtant intéressant d’offrir une architecture qui pourra s’appliquer à l’ensemble d’Internet. Ainsi on ne reste pas limité au réseau RENATER.­­

#### Supervision de l’infrastructure

La supervision de l’architecture sera assurée par un système d’administration bien choisi et adaptée à notre architecture.

Pour tout composant, il est absolument nécessaire de savoir s’ils sont en état de fonctionnement et s’ils répondent aux connexions réseau. Ainsi pour les équipements individuels, un suivi comme suit est réalisé par notre architecture :

* Serveur HTTP

En plus de son accessibilité, nous nous intéresserons à sa charge actuelle ainsi qu’aux utilisateurs qui y sont connectés.

* Poste de supervision (Windows XP)

Le système d’administration doit permettre de superviser le poste de supervision qui, lui, est un système standard avec Windows XP. C’est un poste qui est accessible afin de se connecter aux différentes plateformes via le serveur HTTP.

* PC Vidéo

Le PC vidéo s’occupe des caméras vidéo. Il est donc intéressant de savoir si toutes les caméras sont en état de fonctionnement, éventuellement si elles sont actives (c’est-à-dire qu’un ou plusieurs utilisateurs sont en train de voir son image).

En plus la capacité du disque dur du PC vidéo semble intéressante à superviser, vu que les flux vidéo prennent beaucoup d’espace disque.

* Serveur Linux (DHCP / DNS)

Le serveur Linux héberge les services DHCP et DNS. Le système d’administration doit pouvoir vérifier à tout moment si les deux services sont accessibles.

Pour le service DHCP, le système d’administration permet de savoir quelles adresses IP ont été allouées.

* Connexion VPN avec les autres sites (gérée par le routeur VPN)

Le système de supervision de chaque site doit permettre de visualiser l’état actuel de la connexion VPN. L’état comprend donc la qualité de la liaison (le débit), son utilisation, la durée depuis l’établissement de la connexion.

## Description de la solution

Afin de décrire pleinement le fonctionnement de notre architecture, nous allons détailler ici chaque équipement. Nous décrirons leurs rôles, mais aussi les configurations matérielles et logicielles nécessaires.

#### Routeurs Firewall VPN

Ces routeurs VPN serviront tout d’abord à router les messages à l’intérieur des sous-réseaux. Grâce à leurs fonctionnalités VPN, nous pourront établir un réseau virtuel composé par la juxtaposition des trois sous-réseaux (AIP, GE, IUT Roanne). Enfin, chaque routeur VPN sera un point d’entrée pour le réseau virtuel : ils permettront donc à un tiers de se connecter au VPN et ainsi avoir accès à tous les composants industriels de tous les sites.

Les routeurs devront également intégrer une fonctionnalité de firewall. En effet, les sites se trouvent sur des campus universitaires. Ceux-ci sont normalement assez bien protégés de l’extérieur (Internet) mais le fait d’avoir de nombreux étudiants et chercheurs sur le campus augmente le risque d’une attaque réseau. Ces réseaux ne peuvent donc pas être considérés comme sûrs : c’est la raison pour laquelle nous devons utiliser des firewalls.

Pour assumer tous ces besoins nous avons choisi le routeur firewall **VPN S@N 2000**. Il nous permettra de gérer chaque sous-réseau, son routage, les flux y transitant, et les accès distants sécurisés. Il dispose de 4 ports Ethernet 10/100 Mbits, et sait gérer 128 tunnels VPN en parallèle ce qui suffit à nos besoins.

Ce routeur se trouvera donc en un exemplaire sur chaque site, au plus haut niveau des sous-réseaux. Lors de la configuration de chacun de ces équipements, nous devrons prendre soin de configurer plusieurs éléments :

* Fonctionnalités générales
  + Les fonctionnalités de DHCP intégrées au routeur devront être désactivées : cette fonctionnalité sera assumée par un serveur dédié dans chaque sous-réseau.
* Fonctionnalités VPN
  + Chaque S@N devra être accessible de l’extérieur. Pour garantir ceci, le firewall de chaque campus devra être paramétré pour effectuer une redirection de port vers le S@N. La connexion à chaque VPN se fera donc en utilisant l’adresse IP du routeur ou du firewall du campus et le port alloué à cette redirection.
  + Chaque S@N devra établir 2 tunnels VPN permanents vers les deux autres S@N, afin de créer un réseau virtuel composé par la juxtaposition des 3 sous-réseaux. Les trois sous-réseaux ayant des plages d’adressage différentes, cette juxtaposition se fera sans problème (voir le plan d’adressage).
  + Chaque S@N devra être configuré de manière à permettre les connexions VPN via un client VPN afin de permettre aux élèves et professeurs d’accéder de manière sécurisé au réseau.
* Fonctionnalités Firewall
  + Le firewall doit bloquer toutes les requêtes allant de l’extérieur vers l’intérieur du réseau, ou inversement.
  + Le firewall devra gérer plusieurs profils, établis en fonction des besoins, pour les postes connectés via client VPN. Pour le profil minimum, tous les transferts (même à l’intérieur du réseau) devront être bloqués, mais la connexion au serveur http autorisé. Le profil maximal donnera un plein accès à toutes les ressources du réseau. Les profils seront établis à partir des identifiants fournis lors de la connexion VPN, en se référant au serveur OpenLDAP se trouvant sur le réseau de l’AIP.

#### Switchs

Plusieurs switchs seront présents à différents niveaux de notre architecture. Tout d’abord, chaque site sera équipé d’un switch principal proposant au minimum le découpage en VLAN de niveau 3. Ce switch devra être équipé d’un nombre suffisant de port pour accueillir tous les équipements (au moins un port par plateforme de manipulation). Eventuellement, des switchs plus petits pourront être installés en cascade afin de multiplier le nombre de connexions physiques possibles (voir schéma).

Concrètement, nous choisissons nos équipements parmi le matériel Schneider pour garantir une certaine cohérence avec l’équipement actuel. Nous prenons donc les équipements suivants :

* Site AIP
  + Switch principal : TCS ESM 163F23F0
    - 16 ports
    - VLAN Level 3
    - 10/100 Mbits
  + Switchs supplémentaires (un par plateforme de manipulation) : 499 NES 181 00
    - 8 ports
    - 10/100 Mbits
* Autres sites
  + Switch principal : TCS ESM 083F23F0
    - 8 ports
    - VLAN Level 3
    - 10/100 Mbits

Ces caractéristiques semblent suffisantes, bien que l'on ne connaisse pas exactement le débit utilisé par les plateformes. Nous choisissons un switch 16 ports pour le site principal car il est amené à supporter beaucoup de plateformes de manipulation. Ce switch est administrable de niveau 3. Chaque platine sera également équipé d’un switch 8 ports, cela nous permet donc au total d’installer 12 platines de 8 équipements, plus 4 autres équipements divers, ce qui correspond aux besoins. Ce switch n’est pas administrable. Enfin pour les autres sites, nous installerons un switch 8 ports, car il y a moins d’équipements à brancher, mais administrable de niveau 3.

Les switchs administrables se trouvant sur chaque site vont nous permettre de créer des VLAN. Ceci va nous permettre de segmenter notre réseau VPN en sous-réseau sans tenir compte des contraintes physiques : un VLAN pourra être composé de machines se trouvant sur différents sites.

Nous adoptons la politique de segmentation suivante :

* Chaque manipulation donnera lieu à un VLAN. Ainsi tous les équipements de cette manipulation seront regroupés dans ce VLAN. S’ils diffusent des messages de broadcast, ces-derniers seront diffusés aux autres équipements de la manipulation, mais pas aux autres, qui ne sont pas concernés.
* Le poste de supervision, s’il y en a un sur le site, devra appartenir à tous les VLAN. Ainsi, il pourra superviser l’ensemble de l’installation.

#### Serveur DHCP & DNS

Les serveurs DHCP permettent aux équipements du réseau de recevoir une adresse IP lors de leur démarrage ou de leur branchement sur le réseau. Les serveurs DNS permettent aux équipements réseau d’identifier les différentes machines grâce à des noms.

Les serveurs DNS et DHCP seront confondu sur les mêmes machines, directement branchées à l’interface interne du routeur principal de chaque site. Il s’agira de serveurs de configuration matérielle « basique », car ces serveurs ne seront pas très sollicités. Nous pouvons par exemple choisir un serveur répondant aux caractéristiques suivantes : format Barebone 1U, 1 processeur, 2 Go de RAM.

Ces serveurs exécuteront un OS Linux. Les services suivants devront être lancés :

* DHCPD 4.1.1
* BIND 9.7.0
* OpenSSH 5.4

La configuration détaillée de services DHCP et DNS ainsi que les politiques d’adressage et de nommage seront détaillées dans la partie « Solution pour nommage et l’association des adresses IP ». Le service OpenSSH servira à simplifier l’administration du serveur.

#### Serveur HTTP

Le serveur HTTP met à disposition à distance les plateformes industrielles pour les étudiants ou les enseignants. A travers un navigateur web il est possible de se connecter à celui-ci afin de guider et surveiller les plateformes et équipements industrielles. Ce serveur est déjà en place dans l’architecture actuelle.

Nous situons un serveur HTTP par site, afin de permettre un bon temps de réponse des plateformes aux serveurs. Ils se placeront au niveau du routeur principal et seront accessible dès qu’un utilisateur s’est logué par VPN sur le site respectif. Le serveur HTTP ne sera pas accessible depuis l’extérieur pour des raisons de sécurité. Ainsi, nous imposons aux utilisateurs de lancer une connexion VPN avant de pouvoir utiliser les plateformes industrielles.

En ce qui concerne les caractéristiques de ce serveur, nous choisissons de conserver le serveur existant : nous n’imposons pas de nouveaux besoins spécifiques auxquels celui-ci doit répondre.

#### PC Vidéo

Le PC vidéo sert à gérer les différentes caméras de surveillance des différentes plateformes industrielles. Les caméras vidéo y sont connectées et il envoie les images vidéo vers le serveur Web qui lui, les met à disposition dans son interface web. En plus de ses fonctionnalités le PC vidéo permet de sauvegarder un historique des différents enregistrements vidéo si nécessaire.

Les caractéristiques de ce serveur seront :

* Disque dur : 200 Go
* 1 processeur
* 2 Go de RAM

Au niveau logiciel, nous devrons installer les drivers et logiciels fournis avec les caméras afin de pouvoir mettre les images des caméras à disposition, sur simple connexion IP.

#### Plateformes

Les plateformes resteront inchangées par rapport à la configuration actuelle. Nous devrons juste prendre soin de permettre l’adressage IP via DHCP sur chaque équipement, afin de permettre une gestion dynamique et centralisée de ces adresses.

#### Routeur d’acces de secours

Ce routeur doit permettre de se connecter au site lorsque la connexion principale est rompue. Nous avons choisie d’utiliser un Schneider FSXRAS2000 pour ce propos.

# Solution pour le nommage et l’association des Adresses IP

## Gestion de la politique de nommage

### Definition de la politique

Les équipements du réseau se séparent en deux catégories :

* les éléments des platines de manipulation ;
* les éléments servant à faire fonctionner le réseau ou offrant des services (serveurs, poste de supervision).

Afin de nous retrouver dans le réseau, nous mettons en place un serveur DNS qui permettra d’associer facilement des noms compréhensibles à des adresses IP.

Tous les noms seront composés de différents champs composés de caractères alphanumériques et de tirets uniquement. Ces champs seront séparés par des « \_ ».

Le premier champ correspondra à l’identifiant du site d’exploitation. Ceci nous permettra de situer directement l’équipement.

* Site central : lyon-aip
* Site GE : lyon-ge
* Site Roanne : roanne-iut

Ensuite, s’il s’agit d’un équipement de platine, deux champs suivront :

* Un identifiant de la platine de manipulation, par exemple : bus-can, bus-modbus, etc.
* Un identifiant de l’équipement, composé de manière générique sous la forme « eqp-XX », où XX est un nombre commençant à 01, par exemple : eqp-01, eqp-02, etc.

Ainsi, pour un équipement de platine, le nom DNS pourra ressembler à :

* roanne-iut\_bus-can\_eqp-01
* lyon-aip\_bus-modbus\_eqp-05
* Etc…

Si l’équipement est un élément servant à faire fonctionner le réseau, le deuxième champ contiendra un identifiant permettant de le désigner spécialement.

* Serveurs DHCP & DNS :
  + serv-dhcp-dns
* Serveurs vidéo :
  + serv-video-XX (où XX est un numéro commençant à 01)
* Serveur HTTP :
  + serv-http
* Poste de supervision :
  + poste-supervision

Ainsi, pour un équipement de fonctionnement du réseau, le nom DNS pourra ressembler à :

* lyon-aip\_serv-http
* lyon-ge \_serv-dhcp-dns
* roanne-iut\_poste-supervision
* Etc.

### Organisation du service de DNS multi-domaines

#### Organisation globale et intégration dans les différents campus

Afin de mettre en place la politique de nommage décrite ci-dessus, nous devons mettre en place des serveurs DNS. Pour assurer la meilleure fiabilité du système, nous avons un serveur DNS par site : Ainsi, même si la liaison entre les sites est défaillante, le système pourra fonctionner en local.

Les serveurs DNS (adresses IP 10.0.0.2, 10.0.1.2, etc…) seront placés sur chaque site, juste en-dessous du routeur principal du site (voir schéma global). Ce dernier sera configuré avec le DNS local en tant que DNS primaire et un DNS d’un autre site en secondaire.

#### Organisation des DNS et configuration avec les différentes types d’enregistrements à gérer

Lors de l’ajout d’un équipement sur le réseau, il faudra renseigner plusieurs configurations de la présence de ce nouvel élément. Notamment, il faudra faire l’association entre l’IP de l’équipement et son nom DNS.

Pour cela, nous imposons que les ajouts de configuration se fassent sur le serveur DNS du site principal. Ensuite, comme tous les serveurs devront être configurés à l’identique, un outil d’administration réseau devra permettre de copier facilement, voir automatiquement, le fichier de configuration d’un site à l’autre.

Au niveau du fichier de configuration même, on devra spécifier les enregistrements suivants pour chaque équipement :

**A** – pour faire correspondre un nom d’hôte à une adresse IPv4.

Comme on utilise l’adressage en IPv4, il est évident que ce type d’enregistrement doit figurer dans un serveur terminal.

**PTR –** pour associer une adresse IP à un nom d’hôte (l’inverse du type **A**)

Il nous faut ceci pour pouvoir trouver le nom d’hôte en donnant une adresse IP.

**NS** – définit les serveurs DNS du domaine.

Nous avons 1 serveur DNS principal et des auxiliaires qu’il faut définir, donc forcément ce type d’enregistrement va figurer dans un serveur terminal.

**SOA** – donne les informations générales de la zone : serveur principal, etc.

Pour avoir un résumé des informations générales, ce type d’enregistrement figurera aussi sur le serveur. Ainsi, en faisant une requête de ce type on pourra avoir des informations sur le serveur DNS principal, un courriel de contact, la durée d’expiration, etc.

### Mise en place de l’infrastructure de DHCP

#### Motivation

Afin de permettre le nommage DNS des équipements, nous choisissons de leur attribuer des adresses IP « fixes » que le DHCP leurs assigne: pour un équipement dans un sous-réseau donné, l’adresse IP sera toujours la même. Nous ne pouvons pas nous orienter vers une solution de Dynamic DNS (mise à jour du serveur DNS par le service DHCP). En effet, il faudrait mettre également à jour dynamiquement les VLAN L3, configurés au niveau des switches, et cela n’est pas simplement possible.

Cependant, un même équipement sera potentiellement amené à changer de site, et donc d’adresse IP. Nous ne lui attribuerons donc pas une adresse fixe dans sa propre configuration. L’attribution « fixe » se fera donc via des serveurs DHCP.

#### Gestion des configurations

Il y aura un serveur DHCP par site. Ainsi, même en cas de rupture de liaison entre les sites, les équipements pourront fonctionner en local. Ces serveurs se trouveront sur les même machines que les serveurs DNS, directement sous le routeur principal de chaque site.­­­­

Les serveurs DHCP seront configurés pour donner une adresse IP en fonction de l’adresse MAC de l’équipement : ils reconnaîtront l’équipement et lui donneront son adresse sur le site en question. Chaque serveur DHCP possède un fichier de configuration complet (de tout équipement de tout site).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Site | Adresse Réseau | Masque de sous-réseau | Plage d’adresses IP | Durée du bail | Passerelle par défaut | IP du DNS |
| AIP | 10.0.0.0 | 255.255.255.0 | 10.0.0.0/24 | Long | 10.0.0.1 | 10.0.0.2 |
| GE | 10.0.1.0 | 255.255.255.0 | 10.0.1.0/24 | Long | 10.0.1.1 | 10.0.1.2 |
| Roanne | 10.0.2.0 | 255.255.255.0 | 10.0.2.0/24 | Long | 10.0.2.1 | 10.0.2.2 |

Le serveur DHCP peut détecter de quel sous-réseau vient la demande grâce au relais DHCP. Un agent de relais relaie les paquets DHCP sur le côté distant même s'ils sont des paquets de diffusion. Nous utiliserons donc un tel relais qui fera parti de toutes VLAN du site respectif et qui permet au serveur DHCP de gérer des portées différentes pour les sous-réseaux.

Afin de synchroniser les configurations des DHCP, qui sont identiques pour tous les serveurs DHCP, nous utiliserons les fonctionnalités d’un logiciel d’administration réseau décrit plus loin. Ainsi, nous n’aurons qu’à configurer le serveur DHCP du site principal pour que les autres serveurs soient à jour en quelques minutes seulement, sans intervention humaine.

#### Gestion des adresses IP/MAC

Afin de garantir un réseau cohérent, nous devons définir un plan d’adressage. Nous adoptons la politique suivante :

* Un identifiant unique entre 10 et 254 sera affecté à chaque équipement : ce numéro sera utilisé en octet de poids faible dans l’adresse IP de l’équipement.
* Un identifiant pour identifier le site d’exploitation : ce numéro sera utilisé dans le troisième octet de l’adresse (voir adresses des sous-réseaux sur le schéma).

Exemple : Un équipement avec une adresse MAC 00 :11 :22 :33 :44 :55  aura associé l’identifiant 150. Ensuite sur le sous réseau 10.0.0.0 il aura le serveur DHCP lui associera l’IP 10.0.0.150. Sur le sous-réseau 10.0.1.0 il recevra 10.0.1.150 ainsi de suite.

Au premier branchement d’un équipement, un administrateur réseau choisira le numéro affecté à l’équipement. Il faudra alors mettre à jour la configuration des serveurs DHCP. Il suffira de mettre à jour celle du site principal et les autres serveurs se mettront à jour en quelques minutes seulement, sans intervention humaine, grâce aux fonctionnalités d’un logiciel d’administration réseau décrit plus loin, par réplication du fichier de configuration.

#### Processus de Récupération des adresses MAC

Les adresses MAC peuvent être récupérées de manières différentes :

* Si l’équipement réseau possède une étiquette de sa carte réseau, l’adresse MAC y en inscrit
* Si l’équipement possède un écran et un système d’exploitation il est possible de récupérer son adresse MAC par le biais de la ligne de commande
* Une dernière solution est de se connecter directement par câble réseau à l’équipement avec un ordinateur qui comporte un serveur DHCP et de faire une requête pour obtenir l’adresse MAC.

### Procédure d’ajout de matériel

Nous avons vu que notre architecture nécessitera que quelques interventions soient faites lors de l’ajout d’un nouveau matériel sur le réseau.

Pour récapituler, il faudra :

* Récupérer l’adresse MAC de l’identifiant et y associer un identifiant entre 10 et 254
* Ajouter les réservations d’IP associé à l’adresse MAC dans la configuration DHCP.
* Ajouter les enregistrements DNS associant les noms DNS aux IP.
* Au niveau des switches, configurer à quels VLAN appartient le nouvel équipement (sur chaque switch)

Les configurations DNS et DHCP se répliqueront automatiquement sur tous les serveurs grâce aux outils d’administration mis en place.

# Mise en place d’un système d’administration

## Principe de la solution proposée

D’après les objectifs de la nouvelle architecture, elle doit permettre de superviser le réseau global et permettre une meilleure utilisation des ressources afin de garantir des bonnes performances. En même temps, la solution pour le système d’administration doit permettre à une personne (non spécialiste réseau) l’exploitation quotidienne de l’infrastructure du réseau qui pourra s’appuyer sur du personnel non informaticien pour effectuer des interventions sur les sites distants.

Ainsi, pour pouvoir améliorer et suivre les performances du réseau la solution proposée doit pouvoir faire des mesures de performances pour vérifier la disponibilité du réseau, sa sécurité et sa sûreté. En plus, une surveillance en temps réel du réseau et de ses nœuds devra être proposée pour pouvoir réagir au plus vite en cas de pannes ou des disfonctionnements.

Du côté exploitation, la solution devra inclure des caractéristiques typées (manuel utilisateur, ergonomie adaptée, etc.) pour que la personne en charge puisse effectuer la plupart des manipulations nécessaires en cas de panne (dont le pourcentage en pratique s’élève à plus de 90% de toutes les manipulations effectuées) le plus rapidement et efficacement possible. Il est en même temps clair que la personne en charge devra faire appel à un spécialiste en réseau en l’informant des erreurs ou pannes produites en cas d’incident exceptionnel (moins de 10% de tous les incidents).

En même temps, la solution doit proposer une gestion aisée de la logique et la physique du réseau (auto-détection du nouvel équipement) dans le cas ou la personne en charge doit diriger une personne sur le site distant dans le cadre d’un branchement d’un nouveau matériel ou d’une manipulation.

Ainsi du point de vue organisationnel, la solution proposée doit comprendre :

• Une partie logicielle présente sur chaque poste pour faire des mesures de performances et d’autres indicateurs (occupation espace disque pour les postes vidéo, occupation mémoire pour les postes de supervision, etc.)

• Une partie logicielle de supervision de tout le réseau avec une interface simple (ergonomie adaptée) pour une gestion de l’exploitation quotidienne de l’infrastructure qui doit inclure des messages et des manipulations typées (simples à exécuter) en cas de panne ou erreurs.

Cette partie logicielle doit permettre aussi la gestion des manipulations à distance pour que la personne en charge puisse les effectuer en dirigeant le personnel sur le site distant sans grande difficulté (détection automatique de matériel, reconnaissance et configuration automatique, etc.)

• Une personne sur le site de l’AIP en charge de l’exploitation quotidienne de l’infrastructure du réseau

• Personnel sur les sites distants prêts à faire des manipulations sous la direction de la personne en charge sur le site de l’AIP

• Un manuel utilisateur ciblé pour faire des manipulations à faire en cas de panne ou d’erreur, etc.

Un dernier point, mais pas le moins important, concerne le type des logiciels (gratuit ou payant) à utiliser et leur support après l’achat. La solution doit impliquer une solution logicielle adaptable à la taille du réseau et doit offrir un support client ou une communauté en cas de besoin en évolutivité ou en cas de panne ou tout autre incident exceptionnel.

Etant donne que nous travaillons avec des équipements Schneider, nous sommes emmenés a utiliser soit des logiciels open-source soit des logiciels propriétaires des partenaires de Schneider. Le fait que nous voulons administrer un réseau relativement petit rend les solutions peu couteuses plus intéressantes que les progiciels de monitoring de type HP OpenView, qui sont d’ailleurs dépendants de l’architecture matérielle. Des fonctions telles que la visualisation de la topologie réseau et la découverte automatique deviennent dans ce contexte souhaitables mais non vitales. Dans le cadre de ce projet nous avons donc limite les solutions possibles à deux options générales :

* Utilisation exclusive des logiciels gratuits : Nagios & plugins + MRTG
* La mise en place d’une solution propriétaire : HiVision de Hirschman.

Voici donc l’application de ces solutions a nos besoins :

|  |  |  |  |
| --- | --- | --- | --- |
| Liste besoins | Nagios - NSClient | MRTG | HiVision |
| Notification en cas de pannes - Alarmes |  |  |  |
| Configuration a distance |  |  |  |
| Vérification d’états machines a distance |  |  |  |
| Vérification d’états services a distance |  |  |  |
| Métrologie trafic réseau |  |  |  |
| (Optionnel) Topologie |  |  |  |
| (Optionnel) Découverte automatique |  |  |  |

Les deux solutions couvrent les besoins primordiales que nous avons défini, HiVision couvre même d’autres fonctions optionnelles souhaitables. Tenant compte de la taille réduite du réseau, le cout réduit et la grande communauté derrière l’open-source nous sommes inclines de choisir l’implémentation de Nagios & co. En ce qui concerne le monitoring machines et MRTG en ce qui concerne le trafic réseau. Si l’AIP souhaite une plus grande facilite d’utilisation, l’implémentation de HiVision est également possible, à un cout de 1300€ pour 100 nœuds du réseau.

## Comparaison des différentes solutions logicielles

### Identification de critères de comparaison

Nous avons choisi une liste logique de critères qui découlent des besoins décrits dans la précédente partie et nous avons effectué un tableau de comparaisons des 2 solutions logiciels qu'on prévoit : le couple MRTG & Nagios, ainsi que la solution de Hirschmann – HiVision.

Voici donc nos critères de comparaison :

1. La couverture des fonctions déterminées par les besoins
2. La facilité d'utilisation et de configuration
3. Le type de solution : gratuite ou payante (coût)
4. Évolutivité de la solution
5. Documentation et support

### Tableau récapitulatif

|  |  |  |
| --- | --- | --- |
| Critère | MRTG & Nagios | HiVision |
| La couverture des fonctions déterminées par les besoins | Complète (utilisation de Nagios avec des plugins disponibles) sauf pour les besoins optionnels | Complète |
| La facilité d'utilisation et de configuration | Facilité moyenne - en effet avec Nagios il est plus dure de faire certaines configurations | Bonne |
| Le type de solution : gratuite ou payante (coût) | Totalement gratuite | Payante (700$ - 9000$ en dépendance du nombre de nœuds) |
| Évolutivité de la solution | Moyenne | Bonne |
| Documentation et support | Communauté de support | Support client |

La solution donnée par le couple Nagios et MRTG ne représente par contre pas une configuration qui semble être la mieux adaptée pour une seule personne non informaticienne qui doit gérer l’ensemble du réseau. Nous proposons une solution pour les points non accessibles à l’administrateur non informaticien. Nagios nous présente une interface web assez simple, très riche en informations que nous pensons est adéquate dans notre cas d’utilisation de supervision d’une configuration relativement inchangée. Par contre cette interface ne nous permet pas de surmonter la configuration compliquée à la mise en œuvre du système d’administration ainsi qu’aux changements dans la topologie du réseau. Nous proposons d’automatiser ces changements de configuration à l’aide du développement spécifique. Nous traiterons ces points plus en détail dans le cadre de la « Gestion de la configuration »

## Organisation de la solution

### Besoins en hardware et software

Dans la solution choisie, c.à.d. l’implémentation de Nagios, MRTG et ses plugins nous définissons les besoins suivants :

Besoin Hardware:

* L’existence d’un serveur Linux pour Nagios afin de faire le monitoring par site de chaque réseau local physique.
* Un unique serveur Linux pour Nagios et MRTG dans le réseau VPN qui sert a superviser toutes les machines du système VPN de façon centralisée.

Besoin Software

* Une installation de Nagios sur chacun des serveurs Linux définis aux besoins Hardware
* Pour chaque poste Windows que nous voulons superviser nous allons installer le plugin NSClient++.
* Pour chaque poste Linux que nous voulons superviser nous allons installer le plugin NRPE.
* *En ce qui concerne les équipements Schneider, on nous a dit en séance que Nagios sera compatible mais nous ne savons pas en détail les implications de cette compatibilité. L’internet offre peu d’informations à ce propos.*

### Gestion des configurations

Nous identifions 3 parties de la gestion des configurations. Dans la gestion de la configuration nous identifions trois acteurs importants :

* **L’admin**: personne non informatique avec des connaissances de base de réseau qui gérera l’intégralité du réseau
* **Le geek** : cette personne informaticienne avec des connaissances de réseaux, systèmes d’exploitation et programmation
* **L’aide** : personne qui se trouve sur les différents sites avec des connaissances très limites d’informatique qui aidera l’admin.

**Etant donne qu’en ce qui concerne les systèmes d’administration réseau tout ce qui est utilisable par des non informaticiens coute, nous proposons une solution de gestion qui s’appui pourtant sur l’open source mais qui implique une partie de développement spécifique ponctuel, avant l’installation de la configuration. Le geek s’occupera donc de mettre en place les outils software suivants :**

* Des paquets d’installation et configuration automatique des plugins Nagios. Il développera en script ou langage de programmation des « one click installer » pour les clients NSClient, NRPE, etc.
* Des tutoriels vidéo pour l’installation de Nagios et les paquets.
* Un logiciel simple qui contiendra la topologie réseau (hors ligne) et qui générera les configurations Nagios et clients en fonctions des changements saisis dans un IHM accueillant.
* Développement de plugins Nagios spécialisés pour les plateformes industrielles.
* Développement de scripts permettant de déployer automatiquement les fichiers de configuration de DNS et DHCP entre les sites.

Tenant compte de cela nous proposons que le geek soit un stagiaire par exemple, étudiant IF. Une telle proposition offre une situation avantageuse pour l’étudiant aussi car ce projet semble avoir un fort potentiel pédagogique.

Nous pouvons également envisager que le logiciel de topologie sera une implémentation en tant que plugin Nagios.

Pour résumer, voici le déroulement des trois parties principales de la gestion de configuration :

**L’installation / Mise en œuvre**

* Création des solutions spécifiques a servir a l’admin et aux aides
* Création des tutoriels dédiés pour les aides et l’admin
* Mise en place de l’infrastructure réseau par une personne avise, principalement les routeurs S@N VPN.
* Installation des logiciels par les aides et l’admin.

**L’utilisation des configurations**

* L’admin regardera de temps en temps les graphiques de flux MRTG pour vérifier que le trafic du réseau reste dans les limites nominales.
* L’admin utilisera généralement l’interface web de Nagios pour vérifier l’état des machines réseau. Ces informations seront complémentées par l’image de la topologie du réseau offertes par le logiciel spécifique. Ce type d’utilisation surviendra aux vérifications de maintenance ou au moment des problèmes signalés par les utilisateurs.
* En cas de panne d’un équipement l’admin recevra un e-mail de la part de Nagios qui lui indiquera les problèmes saisies. L’admin essaiera donc de rectifier les problèmes à distance, le cas échéant il demandera de l’aide aux aides pour les taches qui implique l’interaction physique avec les machines.

**La gestion des changements de topologie**

* Nagios est un outil très bien fait pour l’a surveillance des machines fixes dans le réseau mais il nécessite des configurations particulièrement compliques dans le cas ou les machines changent de place dans le réseau ou d’IP. Ces changements devront être saisi manuellement par l’admin dans le logiciel spécifique qui lui rendra en sortie les configurations mises a jour pour Nagios et ses clients. Le logiciel se connectera également par SSH à Nagios et ses clients et va essayer de mettre a jour automatiquement ces fichiers de configuration. Le cas échéant, l’admin sera informe qu’il faut contacter un des aides afin de réaliser l’opération.

### Remarques

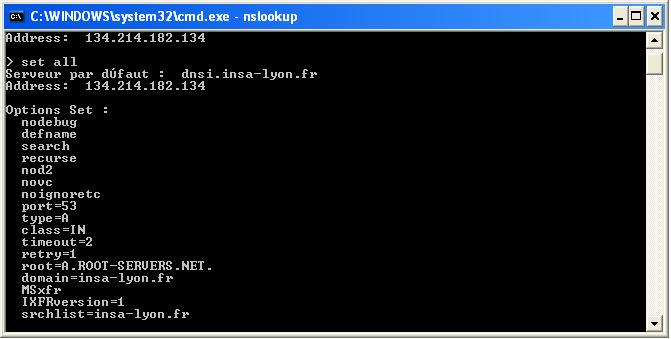
Nous sommes bien conscients que ce chapitre peut ne pas correspondre à vos attentes mais nous avons passe beaucoup de temps essayant de comprendre ce qu’on nous demande sans vraiment de résultat concret. Suite à cela, ce chapitre correspond à ce que nous avons mieux compris. Nous avons pourtant été présents en cours, nous avons utilise le support TP et cours et nous nous sommes même documentes sur wiki et autres communautés en ligne. Nous regrettons que ce chapitre ne reflète pas correctement l’effort déposé dans l’appréhension du sujet.

# Annexe 1 – TEST

## Paramètrage générale

En lançant la commande « nslookup » depuis une fenêtre DOS suivi de la commande « set all » nous arrivons à voir les options du paramétrage, le serveur actuel et l’hôte.

Ainsi l’image suivante affiche le résultat de la commande « set all » dans nôtre cas.

On peut voir que le nom « A.ROOT-SERVERS.NET » est donnée au serveur racine (paramètre « root »). Le nom du serveur de domaine par défaut est mis à « insa-lyon.fr » (paramètre domain) et le port utilisé est le port numéro « 53 ».

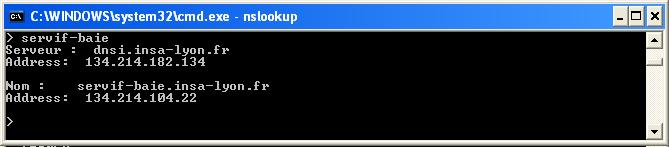
Le paramètre « timeout » fixe la durée d’attente initiale à 2 secondes et le paramètre « retry » fixe à 1 le nombre de tentatives de la requête. Le type de requête est fixé à « A ».

On observe aussi parmi les premiers paramètres qui correspondent à des booléens que :

* On n’affiche pas les informations de débogage (« nodebug »)
* On ajoute le nom de domaine à chaque requête (« defname »)
* On donne une réponse récursive à chaque requête (« recurse »)
* On utilise la liste de recherche de domaine fixée par le paramètre « srchlist » (« search »)
* On n’utilise pas toujours un circuit virtuel (« novc »)
* On utilise le port 53 par défaut pour le DNS (« port »)
* « Type » indique le type DNS, par exemple A pour résoudre une adresse IP à partir d’un nom.
* « Class » spécifie la class utilisé, par exemple IN pour la class Internet
* « timeout » spécifie la durée en seconde d’une attente de réponse
* On utilise le serveur « A.ROOT-SERVERS .NET » comme serveur de DNS de racine
* « domain » indique le domaine par défaut auquel appartient cette machine

## Résolution Servif-baie

Si on essaie de résoudre le nom de servif-baie, on arrive à avoir les informations suivantes :

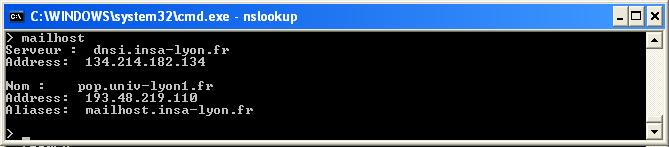


On n’a pas besoin du nom complet car dans les paramètres généraux du DNS on a le nom par défaut du domaine mis à « insa-lyon.fr » (paramètre « domain ») et on ajoute le nom de domaine à chaque requête (paramètre « defname »).

On voit que l’adresse de servif-baie est : 134.214.104.22 et le type d’enregistrement utilisé est « A » (paramètre « type », voir première image).

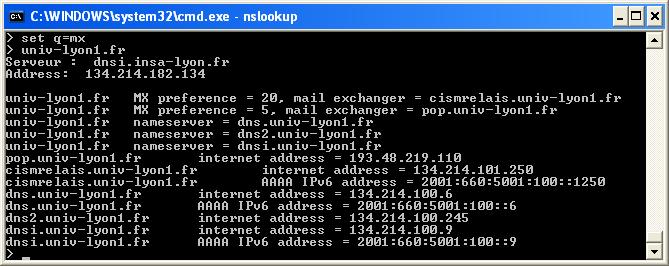
## Résolution Mail

Pour voir le nom de la machine qui héberge le serveur de mail, on exécute la commande « mailhost ».



On voit alors que le nom de la machine qui l’héberge est : « pop.univ-lyon1.fr » et son adresse est : 193.48.219.110.

Pour obtenir l’adresse du serveur de mail de Lyon 1 on exécute les deux commandes suivantes : set « q=mx » (ou set type = mx) et « univ-lyon1.fr ».

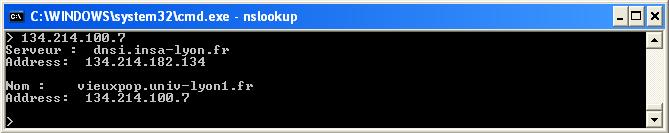


Ainsi on peut voir que Lyon 1 a deux serveurs de mail qui ont une priorité différente.

Le serveur « pop.univ-lyon1.fr » est « préféré » au deuxième serveur (« cismrelais.univ-lyon1.fr »). Ceci peut se voir grâce au numéro précédant l’adresse du serveur. Dans notre cas la préférence de pop.univ-lyon1.fr est égale à 5 donc supérieur à la priorité de cismerelais.univ-lyon1.fr (20).

## Résolution d’un IP

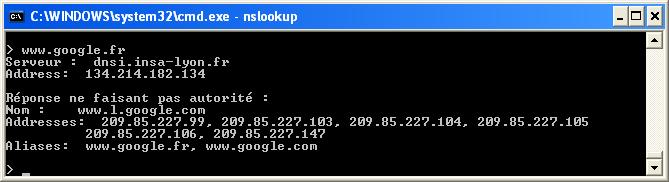
Pour trouver le nom de « 134.214.100.7 » on lance la commande avec cette adresse et on obtient les résultats suivants :



On voit donc que le nom est : vieuxpop.univ-lyon1.fr. Le type d’enregistrement est « A » (on n’oublie pas de resetter le type de la requête à « A » avant l’exécution de cette requête par la commande « set q = A » ou « set type = A »).

## Résolution de www.google.fr

Pour résoudre [www.google.fr](http://www.google.fr) on lance la requête avec le même nom et on obtient les résultats suivants :



Ainsi on voit qu’on est redirigé vers [www.l.google.com](http://www.l.google.com) si on essaie [www.google.fr](http://www.google.fr) ou [www.google.com](http://www.google.com). Nous avons plusieurs adresses car Google utilise plusieurs adresses pour leur gestion de la répartition de la charge. Il y a donc plusieurs adresses (serveurs) auxquels les différentes demandes seront dispatchées.

La réponse ne fait pas l’autorité – cela veut dire que c’est le serveur du domaine parent qui fait autorité et pas celui qui a répondu dans nôtre cas qui correspond juste à un serveur de cache.

Les types d’enregistrement mobilisés sont les suivants :

* « A » : fait correspondre un nom d’hôte à une adresse IPv4.
* « CNAME » : permet de faire d’un domaine un alias vers un autre
* « SOA » : donne les informations générales de la zone : serveur principal, etc. C’est ce type qui permet de nous informer que la réponse ne fait pas autorité.

# Annexe 2 – Monitoring reseau

## Utilisation des commandes de monitoring

### Détermination des paramètres réseaux

Avec la commande *ipconfig* sur notre machine Windows, on trouve :

* Machine Windows 134.214.105.179
* Masque sous-réseau 255.255.255.0
* Passerelle par défaut 134.214.105.1

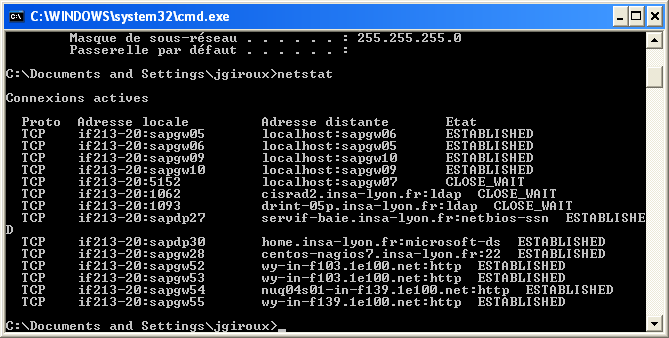
Avec la commande *ifconfig* sur le serveur, on trouve :

* Serveur Nagios 134.214.105.156
* Masque sous-réseau 255.255.255.0

On peut donc en conclure que les deux machines se trouvent sur le même sous-réseau.

### Commande *netstat*

#### *Netstat* sur la machine Windows



Utilisation de la commande de base *netstat*

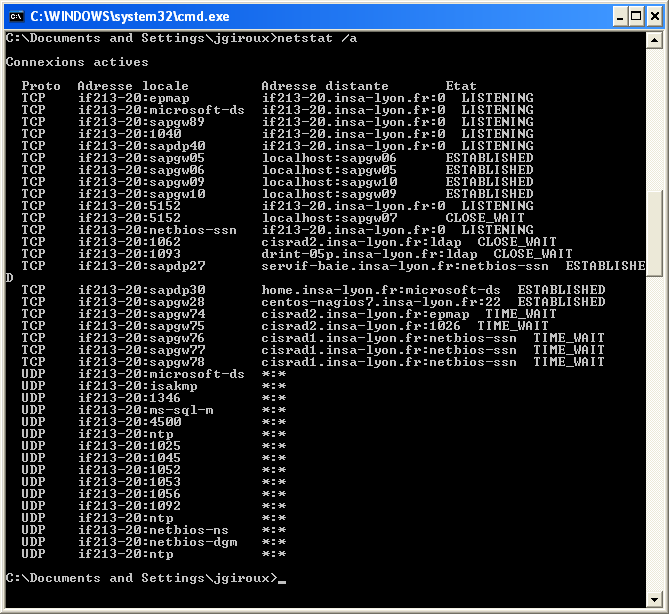
Chaque ligne représente une connexion réseau établie entre la machine locale et une machine distante. Les colonnes correspondent à :

* **Proto** : Le protocole de communication utilisé. Netstat affiche les protocoles TCP, UDP, TCPv6 et UDPv6
* **Adresse locale** : Le nom ou l'adresse IP de la machine sur laquelle la commande netstat est exécutée suivie du numéro de port de sortie de la connexion (séparé par le caractère « : »)
* **Adresse distante** : Le nom ou l'adresse IP de la machine distante avec laquelle la connexion est établie et le port d'entrée de la connexion.
* **Etat** : Classifie l'état de la connexion. Une connexion peut prendre différents états : écoute (LISTENING), établie (ESTABLISHED), en attente (TIME\_WAIT), ou en train de fermeture (CLOSE\_WAIT).

Sous Windows, on peut voir avec la commande de base *netstat* :

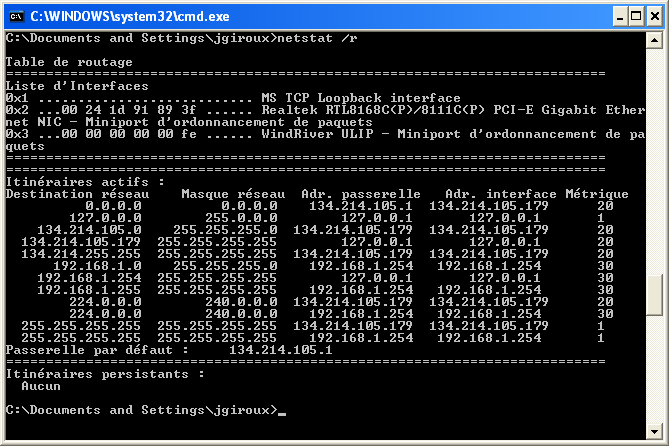
* Les cinq premières connexions sont utilisées par SAPgw (gateway)
* Les quatre connexions suivantes sont avec l’INSA dont les systèmes de fichiers réseau *servif-baie* et *home* (les deux autres sont inexpliquées à notre niveau de connaissance)
* La connexion SSH vers le serveur Nagios
* Les quatre dernières connexions sont avec Google (navigateur ouvert pendant le test)

Par exemple, la première connexion est une connexion TCP à partir de la machine if213-20 sur le port source sapgw05 vers la machine locale sur le port sapgw06. La connexion est en état établie.



Utilisation de la commande *netstat –a* (toutes les connexions)

Avec la commande *netstat -a*, on obtient le même résultat avec en supplément, les connexions qui ne sont pas actives, en attente et les connexions en écoute. Les connexions affichées qui ne présentent pas d’adresse distante (\*:\*) sont en écoute sur le port spécifié dans la colonne « adresse locale ».



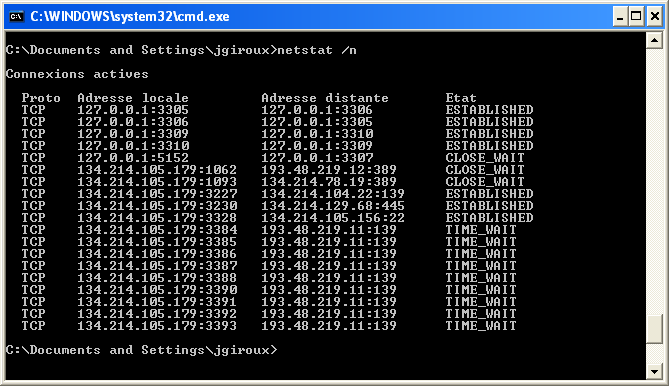
Utilisation de la commande *netstat –r* (table de routage)

Avec la commande *netstat -r*, on obtient la table de routage avec les différents itinéraires actifs. A chaque destination figure son adresse réseau, son masque sous-réseau, sa passerelle, son interface et sa métrique. On remarquera notamment les adresses :

* 127.0.0.0 localhost (loopback)
* 134.214.105.179 la machine elle-même
* 134.214.105.255 adresse de broadcast du réseau
* 192.168.1.0 adresse de réseau privé

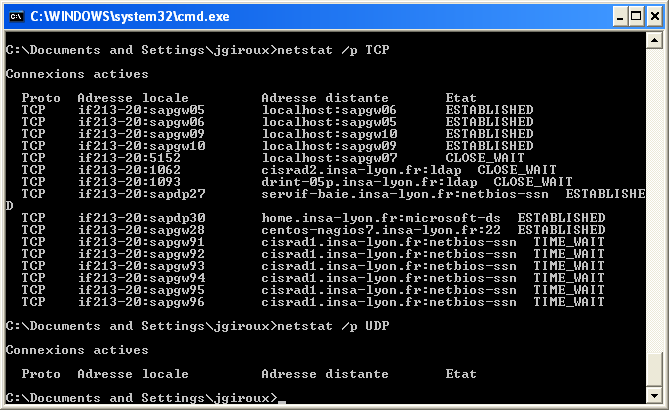
Par exemple, si on a une connexion à faire vers l’adresse 134.214.105.42/24, l’ordinateur va voir que cette adresse fait partie de son propre sous-réseau et se connecter directement.

Si on cherche à faire une connexion vers iftpserv2 (134.214.104.18/24), on se rend compte qu’il n’y a aucune règle de comportement dans la table de routage concernant ce réseau. On envoie alors vers la passerelle par défaut qui est 134.214.105.1.



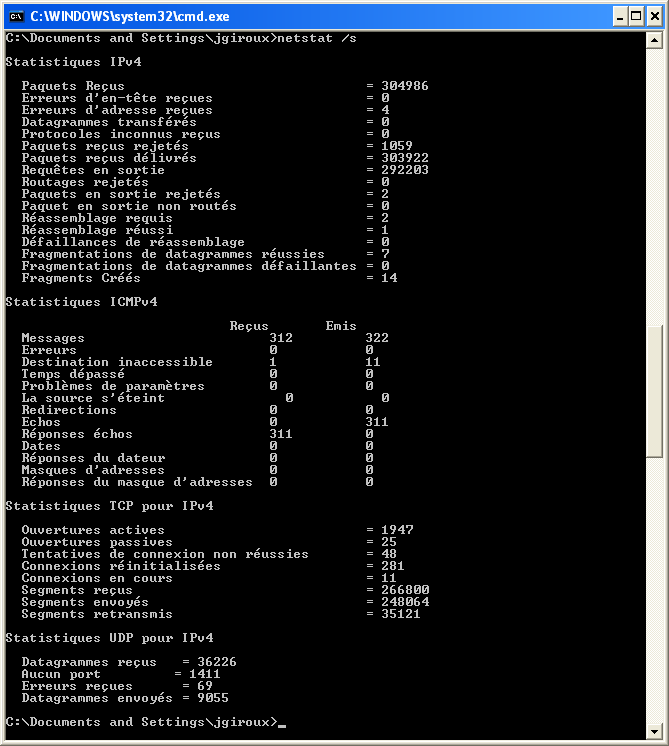
Utilisation de la commande *netstat –n* (adresses et numéro de ports actifs)

Avec la commande *netstat -n*, on remarque que seules sont actives les connexions utilisant l’adresse de loopback et du réseau classique. Celles-ci ne fonctionnent d’ailleurs qu’avec le protocole TCP.

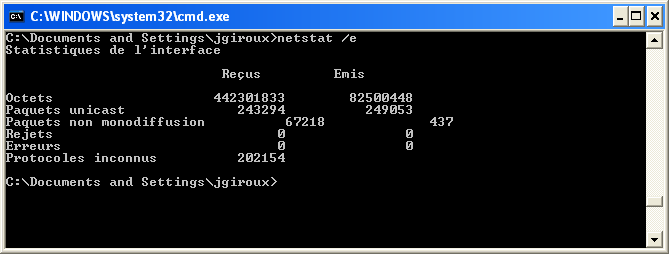


Utilisation de la commande *netstat –p proto* (connexions actives pour le protocole *proto*)

On confirme le résultat vu à l’image précédente puisque l’on voit que, si on affiche les connexions actives par protocole, aucune connexion UDP n’est active.



Utilisation de la commande *netstat –s* (statistiques par protocole)

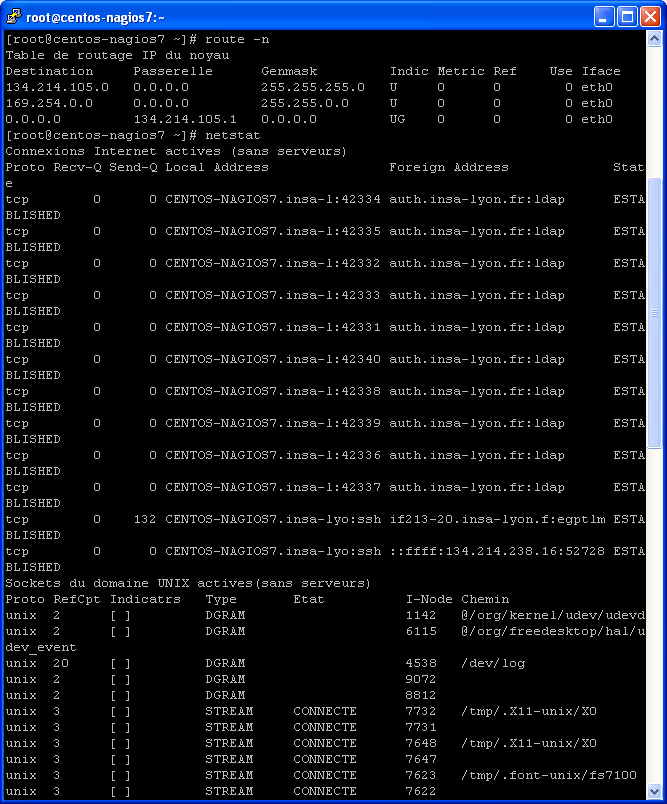


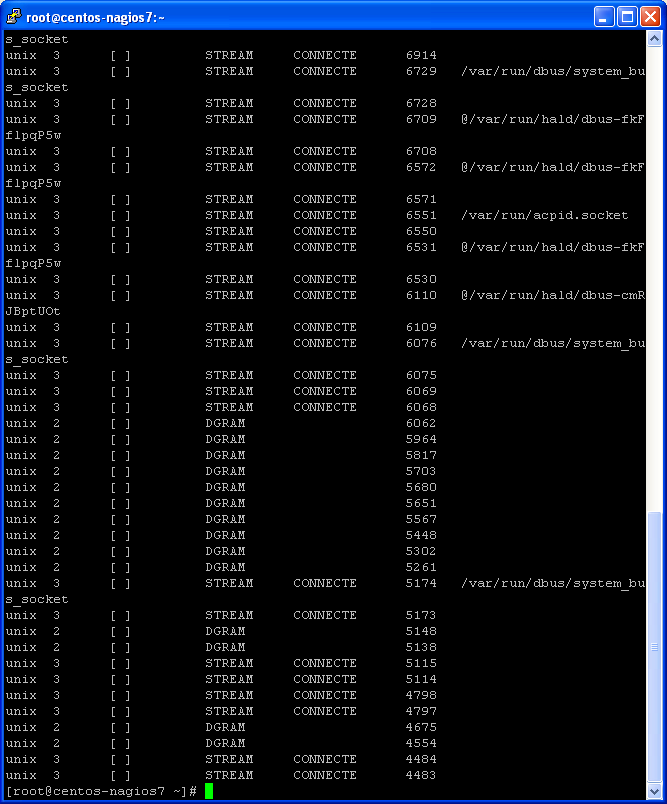
Utilisation de la commande *netstat –e* (statistiques Ethernet)

Enfin, les commandes *netstat –s* et *netstat –e* permet d’afficher les informations sur les statistiques des connexions et les statistiques des paquets qu’elles permettent d’échanger.

Pour la suite, on va appliquer les mêmes commandes à partir du serveur nagios par connexion SSH.

#### *Netstat* sur le serveur Nagios



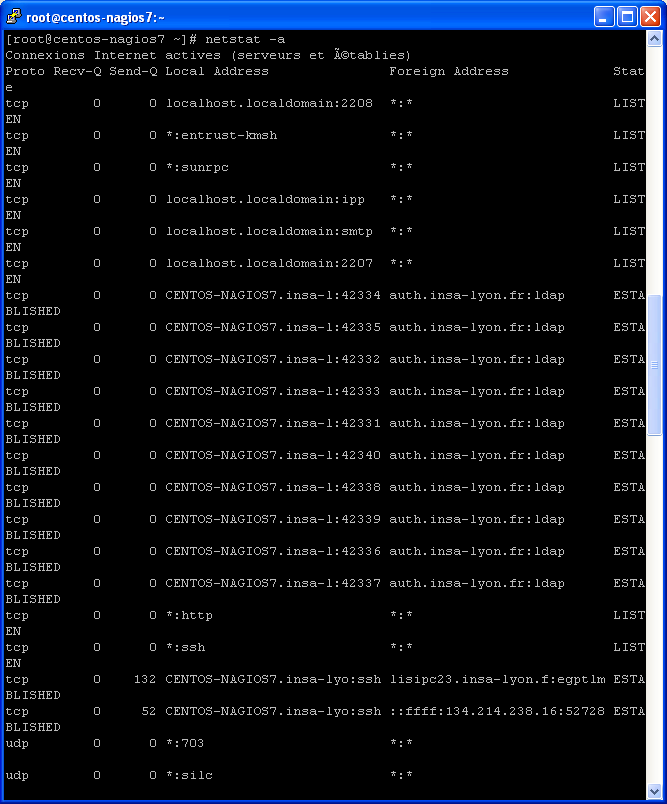


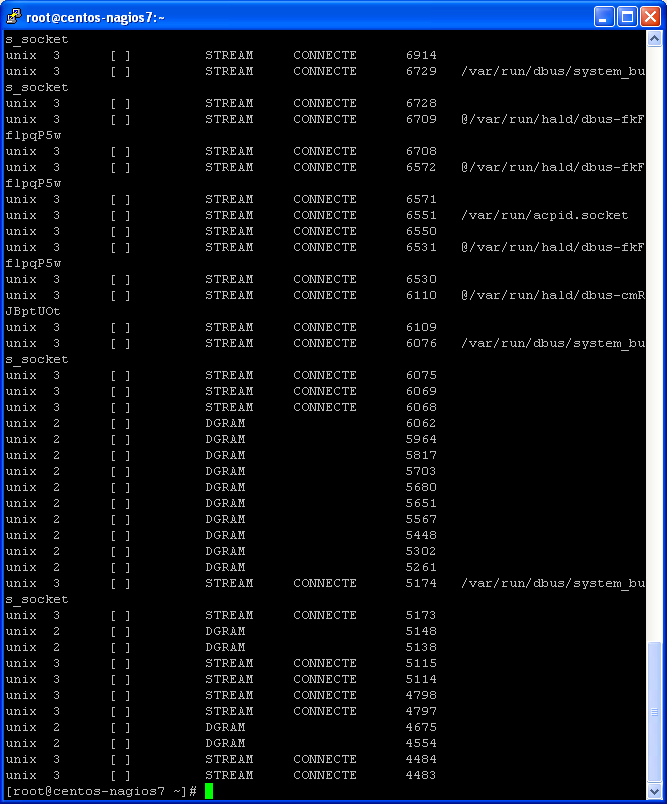
Utilisation de la commande de base *netstat*

L’affichage se décompose en 2 parties : l’une sur les connexions internet actives et l’autre sur les sockets actifs.

La première partie reprend le même affichage que sur machine Windows (à l’exception que l’on se trouve sous Unix) ; pour une explication, reportez vous à la partie netstat sous Windows présentée précédemment.

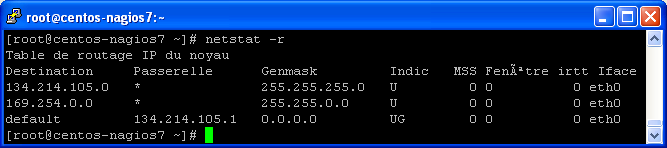
La deuxième partie présente les sockets actifs. Cette partie n’est pas utile pour notre étude du réseau (notamment au niveau architecture du réseau). On notera tout de même la nature des échanges dans la colonne « Type » : DGRAM pour datagram (UDP) ou STREAM (TCP)





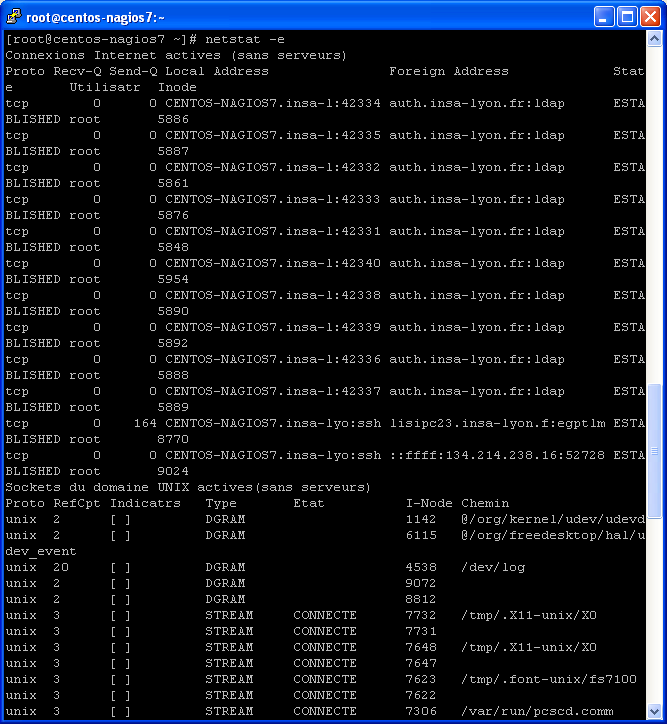
Utilisation de la commande *netstat –a*

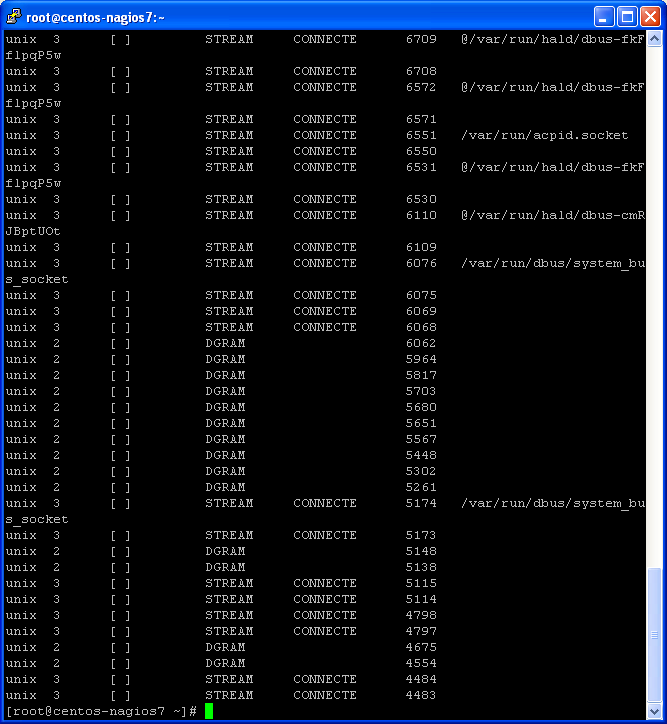
De la même manière que sous Windows, on affiche l’ensemble des connexions. On remarque encore une fois que les connexions présentant le symbole « \*:\* » font de l’écoute de port (port indiqué dans la colonne « Local address »).



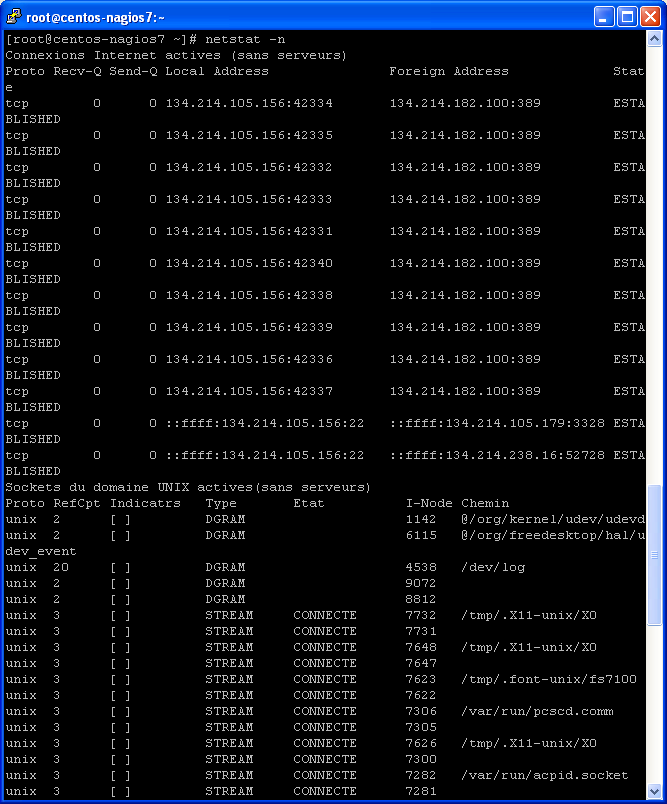
Utilisation de la commande *netstat -r*

La table de routage, bien que plus simple, présente la même forme. On retrouve d’ailleurs la destination réseau 134.214.105.0/24 et la passerelle par défaut 134.214.105.1.



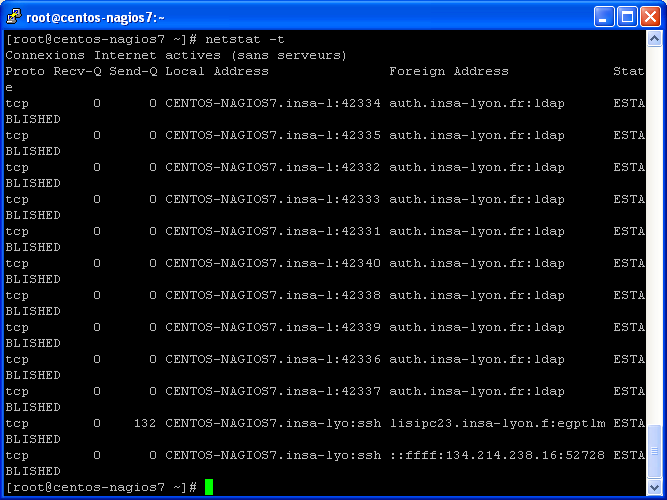


Utilisation de la commande *netstat –e* (extended)



Utilisation de la commande *netstat –n* (numeric)

Les adresses IP ne sont pas remplacées par leur nom.



Utilisation de la commande *netstat –t* (équivalent *–p TCP*)

Idem à la commande Windows

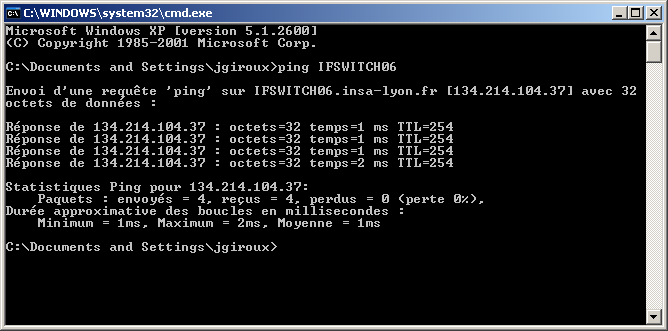


Utilisation de la commande *netstat –u* (équivalent *–p UDP*)

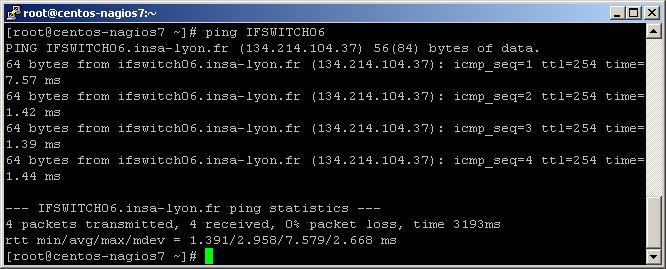
Idem à la commande Windows

## Sniffer de paquet et ping sur IFSWITCH06

Ping est un utilitaire réseau, utilisé pour déterminer si une machine est accessible sur le réseau. Cette section s'intéresse au ping de la machine IFSWITCH06. Dans un premier temps, nous lançons un ping depuis la machine Windows et depuis une autre machine connectée en SSH.



Ping à partir de la machine Windows



Ping à partir du serveur Nagios

Nous voyons que IFSWITCH06 est accessible par les deux machines. (ping nous donne également quelques informations comme le temps de réponse et l'ip de la machine)

Nous analysons maintenant les paquets utilisés par ping. Cette analyse sera faite à partir du poste Windows car l'outil que nous allons utiliser (Wireshark) est plus simple d'utilisation que les commandes Unix. Wireshark est un sniffer de paquets réseau. Il nous permet de capturer les paquets utilisés par ping. Voici les deux paquets échangés lors d'un ping unitaire :

No. Time Source Destination Protocol Info

844 24.144930 134.214.105.179 134.214.104.37 ICMP Echo (ping) request

Frame 844 (74 bytes on wire, 74 bytes captured)

Arrival Time: Mar 18, 2010 15:15:56.157076000

[Time delta from previous captured frame: 0.003080000 seconds]

[Time delta from previous displayed frame: 24.144930000 seconds]

[Time since reference or first frame: 24.144930000 seconds]

Frame Number: 844

Frame Length: 74 bytes

Capture Length: 74 bytes

[Frame is marked: True]

[Protocols in frame: eth:ip:icmp:data]

[Coloring Rule Name: ICMP]

[Coloring Rule String: icmp || icmpv6]

Ethernet II, Src: Giga-Byt\_91:89:3f (00:24:1d:91:89:3f), Dst: 3com\_46:0d:81 (00:1a:c1:46:0d:81)

Destination: 3com\_46:0d:81 (00:1a:c1:46:0d:81)

Address: 3com\_46:0d:81 (00:1a:c1:46:0d:81)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Source: Giga-Byt\_91:89:3f (00:24:1d:91:89:3f)

Address: Giga-Byt\_91:89:3f (00:24:1d:91:89:3f)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Type: IP (0x0800)

Internet Protocol, Src: 134.214.105.179 (134.214.105.179), Dst: 134.214.104.37 (134.214.104.37)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)

0000 00.. = Differentiated Services Codepoint: Default (0x00)

.... ..0. = ECN-Capable Transport (ECT): 0

.... ...0 = ECN-CE: 0

Total Length: 60

Identification: 0xcff6 (53238)

Flags: 0x00

0.. = Reserved bit: Not Set

.0. = Don't fragment: Not Set

..0 = More fragments: Not Set

Fragment offset: 0

Time to live: 128

Protocol: ICMP (0x01)

Header checksum: 0x8b45 [correct]

[Good: True]

[Bad : False]

Source: 134.214.105.179 (134.214.105.179)

Destination: 134.214.104.37 (134.214.104.37)

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0 ()

Checksum: 0x2958 [correct]

Identifier: 0x0200

Sequence number: 8708 (0x2204)

Data (32 bytes)

Data: 6162636465666768696A6B6C6D6E6F707172737475767761...

[Length: 32]

0000 00 1a c1 46 0d 81 00 24 1d 91 89 3f 08 00 45 00 ...F...$...?..E.

0010 00 3c cf f6 00 00 80 01 8b 45 86 d6 69 b3 86 d6 .<.......E..i...

0020 68 25 08 00 29 58 02 00 22 04 61 62 63 64 65 66 h%..)X..".abcdef

0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv

Paquet *echo request* : grâce à Wireshark on obtient une analyse complète et très détaillée du paquet réseau et de la signification de chacun de ses bits. On peut par exemple voir que le paquet pèse 74 octets, les adresses MAC de la source et de la destination, etc.

Intéressons nous au coeur de ping : on voit que le protocole ICMP est utilisé. Ce protocole réside dans IP au niveau 3, c'est-à-dire qu'il est directement donné par une interface réseau, ce n'est pas un protocole applicatif (on peut par exemple voir qu'il n'y a pas de notions de numéro de port). Parmi les champs de ICMP, on retiendra le type (8 signifie un requête d'*echo*) et une plage de données utilisée pour vérifier la fiabilité du réseau.

No. Time Source Destination Protocol Info

392 5.756208 134.214.104.37 134.214.105.179 ICMP Echo (ping) reply

Frame 392 (74 bytes on wire, 74 bytes captured)

Arrival Time: Mar 18, 2010 15:51:59.910434000

[Time delta from previous captured frame: 0.001203000 seconds]

[Time delta from previous displayed frame: 0.001203000 seconds]

[Time since reference or first frame: 5.756208000 seconds]

Frame Number: 392

Frame Length: 74 bytes

Capture Length: 74 bytes

[Frame is marked: True]

[Protocols in frame: eth:ip:icmp:data]

[Coloring Rule Name: ICMP]

[Coloring Rule String: icmp || icmpv6]

Ethernet II, Src: 3com\_46:0d:81 (00:1a:c1:46:0d:81), Dst: Giga-Byt\_91:89:3f (00:24:1d:91:89:3f)

Destination: Giga-Byt\_91:89:3f (00:24:1d:91:89:3f)

Address: Giga-Byt\_91:89:3f (00:24:1d:91:89:3f)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Source: 3com\_46:0d:81 (00:1a:c1:46:0d:81)

Address: 3com\_46:0d:81 (00:1a:c1:46:0d:81)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Type: IP (0x0800)

Internet Protocol, Src: 134.214.104.37 (134.214.104.37), Dst: 134.214.105.179 (134.214.105.179)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)

0000 00.. = Differentiated Services Codepoint: Default (0x00)

.... ..0. = ECN-Capable Transport (ECT): 0

.... ...0 = ECN-CE: 0

Total Length: 60

Identification: 0x4ac4 (19140)

Flags: 0x00

0.. = Reserved bit: Not Set

.0. = Don't fragment: Not Set

..0 = More fragments: Not Set

Fragment offset: 0

Time to live: 254

Protocol: ICMP (0x01)

Header checksum: 0x9277 [correct]

[Good: True]

[Bad : False]

Source: 134.214.104.37 (134.214.104.37)

Destination: 134.214.105.179 (134.214.105.179)

Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0 ()

Checksum: 0x1f58 [correct]

Identifier: 0x0200

Sequence number: 13316 (0x3404)

Data (32 bytes)

Data: 6162636465666768696A6B6C6D6E6F707172737475767761...

[Length: 32]

0000 00 24 1d 91 89 3f 00 1a c1 46 0d 81 08 00 45 00 .$...?...F....E.

0010 00 3c 4a c4 00 00 fe 01 92 77 86 d6 68 25 86 d6 .<J......w..h%..

0020 69 b3 00 00 1f 58 02 00 34 04 61 62 63 64 65 66 i....X..4.abcdef

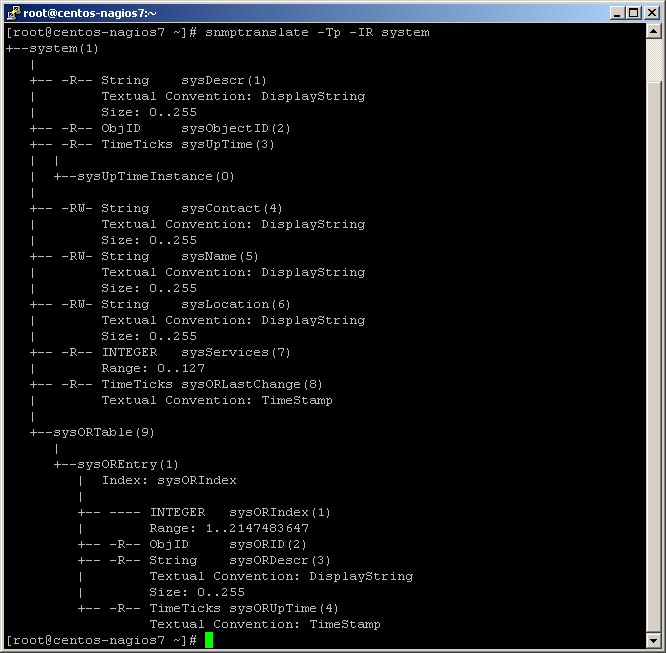
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmnopqrstuv

0040 77 61 62 63 64 65 66 67 68 69 wabcdefghi

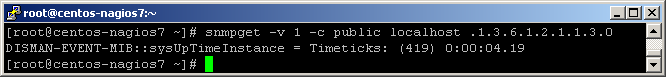
Paquet *echo* *reply* : on voit cette fois-ci que le type ICMP est 0 (réponse à la requête) et que les données ont bien été conservées.

## Serveur nagios et MiB

Dans cette section nous nous intéressons à la supervision de machines grâce au protocole SNMP et plus précisément aux informations contenues dans la MIB d'un machine avec service Nagios. La MIB est une base de données stockant les informations relatives au matériel sur lequel elle est installée. Ces informations sont nécessaires pour manager un réseau car elles fournissent des informations pratiques sur la localisation géographique et structurelle du matériel, sur son fonctionnement et sa charge de travail, etc.

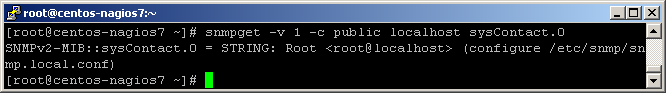


Affichage de la MIB de la machine : on voit que les informations sont stockées sous la forme d'un arbre. Certaines informations sont modifiables (-RW- comme sysContact, sysName...) et d'autres sont en lecture seule (-R-- comme sysDescr...).



L’uptime indique la durée depuis la mise en fonctionnement du service. Ici, il est en fonctionnement depuis un peu plus de 4 secondes (normal on vient de le lancer). L'uptime peut être utile à savoir, pour par exemple voir s'il y a eu des coupures de courant, etc.

Note : on voit dans la fin du chemin que l'information accédée est sysUpTimeInstance dans l'arbre précédemment montré.



Ici, on obtient le sysContact de la MiB (chemin équivalent à .1.3.6.1.2.1.1.4.0).

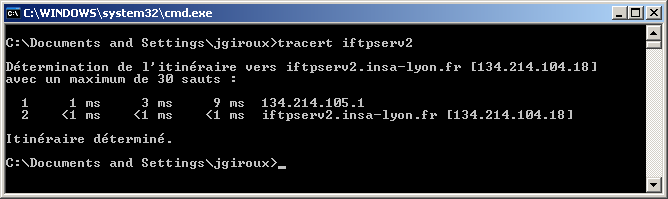
## Topologie par *traceroute*

Traceroute (ou tracert) est un utilitaire réseau utilisé pour connaître la route empruntée par des paquets sur le réseau.

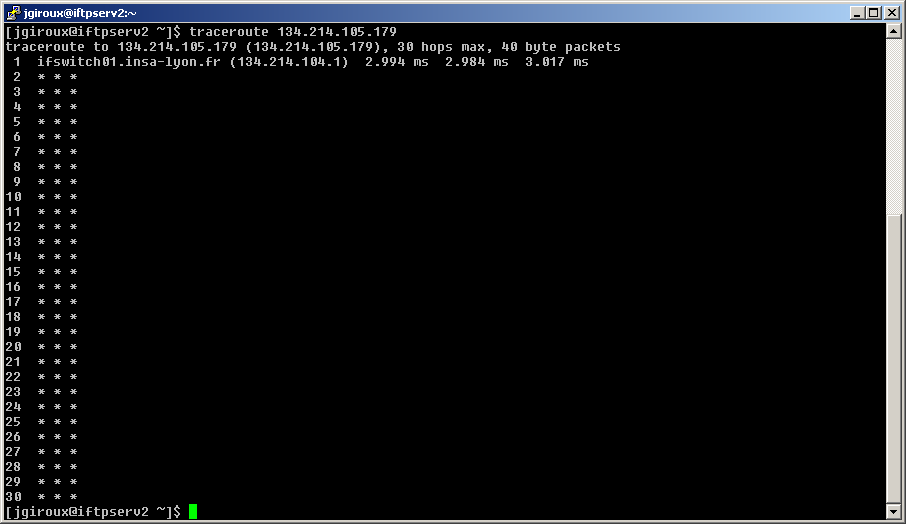
Pour expliquer le principe de fonctionnement, il faut d'abord connaître la signification du champ TTL d'un paquet IP. Le champ TTL (Time to live) d'un paquet détermine le nombre maximum de transmissions de ce paquet. Le paquet est envoyé avec une certaine valeur TTL. Chaque machine que le paquet rencontre sur sa route diminue le TTL de 1, jusqu'à ce que ce champ soit nul. A ce moment-là, il devient ignoré et n'est plus retransmis. Un message ICMP est renvoyé à l'émetteur.

Il est maintenant facile de comprendre comment fonctionne l'utilitaire traceroute : trois paquets avec un TTL de 1 sont envoyés, puis de 2, etc. On obtient donc une liste progressive des machines rencontrées sur la route des paquets.

Dans cette section nous lançons un traceroute de iftpserv2 vers la machine de TP et inversement, et en déduisons la topologie du réseau.



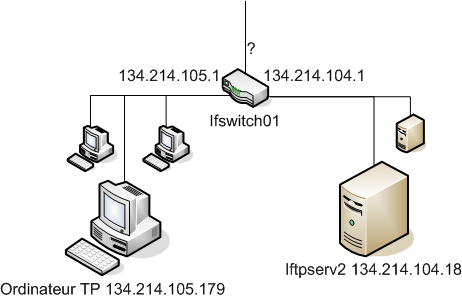
Commande côté machine Windows vers iftpserv2 : on voit que le paquet passe par la passerelle de la machine puis attend directement la cible. On obtient les temps entre l'émission du paquet et la réception de la commande ICMP pour chacun des trois paquets.



Commande côté iftpserv2 vers la machine Windows : on voit que le paquet passe par ifswitch01, puis ne répond plus (\* signifie un time out pour un paquet).

On en déduit que la passerelle de la machine Windows et ifswitch01 sont en fait le même routeur, avec des interfaces différentes. La politique de sécurité au moment du test était de ne pas relayer les messages ICMP en provenance du sous-réseau de la machine Windows, ce qui explique les \* \* \*.

On peut en déduire une topologie de la forme suivante :



# Annexe 3 – NAGIOS

## Introduction

Dans le cadre de cette partie nous avons regardé comment mettre en place un système de monitoring à distance d’une machine Windows. Pour réaliser cela, nous avons utilisé le couple Nagios - NSClient++. Ces deux logiciels open source représentent une solution relativement simple et efficace pour le monitoring des machines d’un réseau.

## Remarques sur le tutoriel d’installation

* Il semble être plus facile d’accéder aux fichiers .cfg par l’intermédiaire d’un logiciel comme *WinSCP* ou *Filezilla* que de les éditer par *vi* sous *Putty*.
* Pour rajouter plusieurs machines Windows à Nagios, nous avons choisi la méthode suivante :
  + Créer un fichier .cfg par machine (ex : *myLaptopWindows.cfg, pcIF.cfg*)
  + Garder une **unique** définition du host group dans un seul de ces fichiers.
  + Bien vérifier qu’en *nagios.cfg* tous les .cfg par machine sont mentionnés comme fichiers de configuration.
* La validation de ces fichiers de configuration devra se faire avec ce message très convivial : "Things look okay - no problems were detected during the pre-flight check"
* Suite au fait qu’à la fin de l’installation, Nagios nous montrait « Explorer.exe » comme « CRITICAL » sur une machine Windows 7, nous avons rajouté une autre machine Windows XP du département pour voir si c’est un problème de compatibilité entre NSClient++ et Windows. Ce n’était pas le cas, nous avons juste mal interprété cette donnée qui semble être valide puisqu’Explorer n’était pas utilisé au moment des vérifications.

## Fichiers de configuration

Pour des raisons écologiques et de lisibilité, nous allons vous présenter les fichiers de configuration sans commentaires ou partiellement. Les versions électroniques se retrouvent sur le serveur **centos-nagios7** auquel vous avez sûrement accès.

### NSC.ini

Voici le fichier de configuration / initialisation de NSClient++ dont les commentaires ont été enlevés pour le rendre plus lisible. Ce fichier est identique pour les deux machines que l’on suit puisqu’il décrit l’unique serveur Nagios aux NSCLients.

[modules]

FileLogger.dll

CheckSystem.dll

CheckDisk.dll

NSClientListener.dll

NRPEListener.dll

SysTray.dll

CheckEventLog.dll

CheckHelpers.dll

[Settings]

password=passé

allowed\_hosts=134.214.105.156

[log]

[NSClient]

port=12489

[Check System]

[NRPE]

[NRPE Handlers]

### Nagios.cfg

La seule modification de ce fichier a été de rajouter les fichiers de configuration pour chaque machine Windows :

# Definitions for monitoring a Windows machine

cfg\_file=/usr/local/nagios/etc/objects/windows.cfg

cfg\_file=/usr/local/nagios/etc/objects/windowsIF.cfg

### windows.cfg

Ceci est le fichier de configuration pour un portable sous Windows 7 : myLaptopServ. Les commentaires ont été enlevés.

define host{

use windows-server ; Inherit default values from a template

host\_name myLaptopServ ; The name we're giving to this host

alias Ze Windowz laptop server; A longer name associated with the host

address 134.214.239.21 ; IP address of the host

}

define hostgroup{

hostgroup\_name windows-servers ; The name of the hostgroup

alias Windows Servers ; Long name of the group

}

define service{

use generic-service

host\_name myLaptopServ

service\_description NSClient++ Version

check\_command check\_nt!CLIENTVERSION

}

define service{

use generic-service

host\_name myLaptopServ

service\_description Uptime

check\_command check\_nt!UPTIME

}

define service{

use generic-service

host\_name myLaptopServ

service\_description CPU Load

check\_command check\_nt!CPULOAD!-l 5,80,90

}

define service{

use generic-service

host\_name myLaptopServ

service\_description Memory Usage

check\_command check\_nt!MEMUSE!-w 80 -c 90

}

define service{

use generic-service

host\_name myLaptopServ

service\_description C:\ Drive Space

check\_command check\_nt!USEDDISKSPACE!-l c -w 80 -c 90

}

define service{

use generic-service

host\_name myLaptopServ

service\_description W3SVC

check\_command check\_nt!SERVICESTATE!-d SHOWALL -l W3SVC

}

define service{

use generic-service

host\_name myLaptopServ

service\_description Explorer

check\_command check\_nt!PROCSTATE!-d SHOWALL -l Explorer.exe

}

### windowsIF.cfg

Ceci est le fichier de configuration pour une machine du département IF sous Windows XP : ifMachine. Les commentaires ont été enlevés. A remarquer l’absence de  définition de host group faite auparavant dans windows.cfg.

define host{

use windows-server ; Inherit default values from a template

host\_name ifMachine ; The name we're giving to this host

alias Ze Windows IF PC server; A longer name associated with the host

address 134.214.105.177 ; IP address of the host

}

define service{

use generic-service

host\_name ifMachine

service\_description NSClient++ Version

check\_command check\_nt!CLIENTVERSION

}

define service{

use generic-service

host\_name ifMachine

service\_description Uptime

check\_command check\_nt!UPTIME

}

define service{

use generic-service

host\_name ifMachine

service\_description CPU Load

check\_command check\_nt!CPULOAD!-l 5,80,90

}

define service{

use generic-service

host\_name ifMachine

service\_description Memory Usage

check\_command check\_nt!MEMUSE!-w 80 -c 90

}

define service{

use generic-service

host\_name ifMachine

service\_description C:\ Drive Space

check\_command check\_nt!USEDDISKSPACE!-l c -w 80 -c 90

}

define service{

use generic-service

host\_name ifMachine

service\_description W3SVC

check\_command check\_nt!SERVICESTATE!-d SHOWALL -l W3SVC

}

define service{

use generic-service

host\_name ifMachine

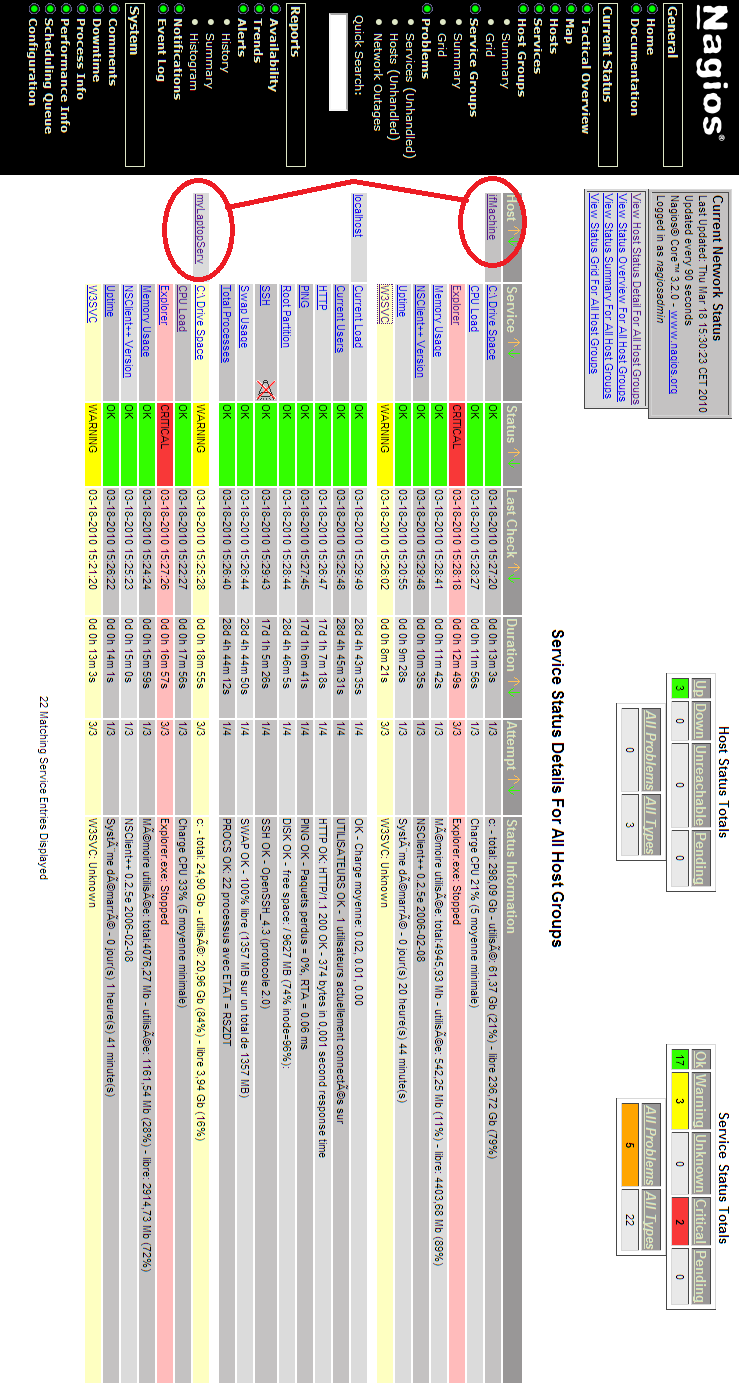
service\_description Explorer

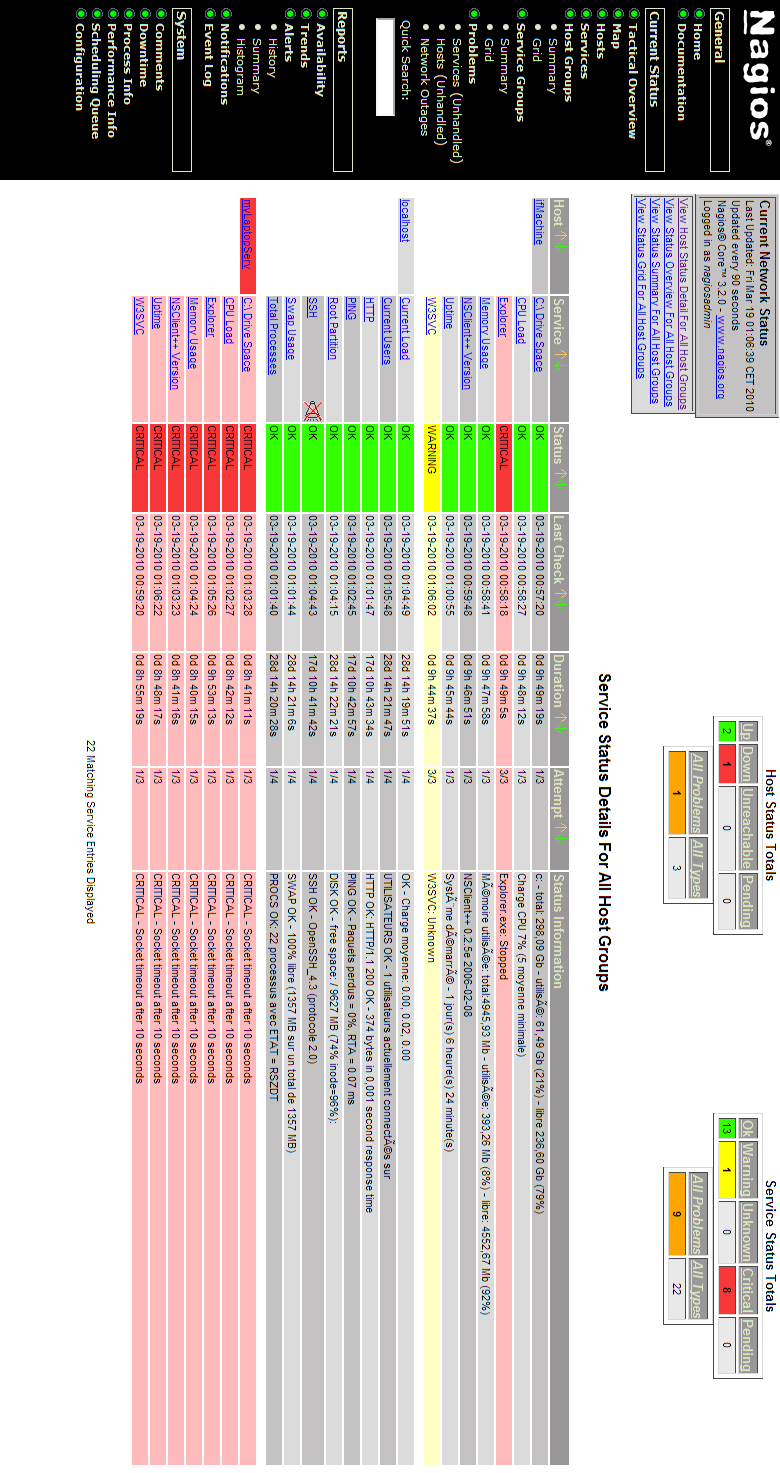
check\_command check\_nt!PROCSTATE!-d SHOWALL -l Explorer.exe

}

## Captures d’ecran

Nous vous présentons deux captures d’écran. Elles présentent les trois machines surveillées par Nagios : sa propre machine *localhost*, un portable *myLaptopServ* et un PC de l’IF *ifMachine*. Dans la deuxième image, le portable est éteint.





## Analyse critique

### Sur l’installation

L’installation du client NSClient++ ainsi que la configuration du serveur Nagios semble assez simple en suivant le tutoriel mais elle n’est pas forcément intuitive sans documentation ou aide. Ceci est le cas pour la plupart des logiciels open-source qui ne sont pas facilement compréhensibles pour la majorité des utilisateurs. Heureusement, Nagios et NSClient sont largement utilisés, et donc soutenus par une communauté et beaucoup de documents et de tutoriels d’utilisation.

Du point de vue d’un étudiant ingénieur en IF, les fichiers de configuration sont assez clairement écrits et nous comprenons vite comment les modifier en fonction de ce dont nous avons besoin. L’architecture de Nagios par modules et services simplifie beaucoup le processus de compréhension.

### Sur l’utilisation

En accédant à Nagios par son portail web nous entrons tout de suite dans le monde des interfaces conviviales et nous comprenons vite à quoi il peut servir. Tous les services de NSClient++ définis dans les .cfg des machines (windows.cfg, windowsIF.cfg) apparaissent regroupés par hôte et nous montrent assez clairement l’état courant de l’hôte. Les requêtes se font assez souvent pour avoir un suivi « temps réel » pour faire de la bonne administration, des vérifications manuelles sont également possibles.

Nagios permet de regrouper les états de plusieurs types d’objets réseau :

* Imprimantes
* Machines Windows et linux
* Routeurs et Switches

Il représente un outil puissant pour voir **l’état des équipements** mais il ne contient aucun module de dessin de la topologie du réseau. Il devient alors très bien adapté pour avertir le dysfonctionnement d’un équipement ou faire des statistiques sur certains points du réseau mais il peut y avoir certains problèmes qui ne sont pas évidents à dépister, et où un outil qui connait la topologie du réseau sera beaucoup plus adapté (nous pouvons imaginer des problèmes d’optimisation de la bande passante, d’isolation des surcharges multi-cast, etc).

En effet, plusieurs situations peuvent compliquer le monitoring réseau par Nagios :

* La gestion des machines avec IP dynamique est impossible à cause des fichiers de configuration qui demandent l’IP et non le nom de l’hôte.
* Pour une poignée de machines, Nagios peut être suffisant pour faire de l’administration mais pour des réseaux plus importants, voire plus de 100 machines sous forme d’une liste n’est peut-être pas la meilleure des solutions même si le regroupement en Host Groups est possible.

Pour avoir une solution plus complète en ce qui concerne l’administration réseau, nous pourrions :

* Utiliser des outils propriétaires payants : HP OpenView, HiVision
* Installer gratuitement Nagios et des clients, avoir une carte de la topologie et beaucoup d’imagination…

Utiliser l’avantage open source et développer un module Nagios qui lit les MiBs pour découvrir la topologie réseau et faire des meilleures statistiques.

# Annexe 4 – Interpretation des Courbes de MRTG

## « Weekly » Graph (30 minute average)

Ce graphe qui visualise le trafic d’une semaine donné permet de saisir les informations suivantes :

* Une surcharge pendant les jours 6, 7 et 8. Cette surcharge
* La valeur du trafic est nulle entre 23h du soir et 7h du matin. Cette coupure correspond probablement à une coupure de courant et nécessite donc une intervention humaine.

## « Daily » Graph (5 minute average)

Le trafic analysé par ce graphe présente le trafic d’une journée. On constate que même pendant la nuit le trafic reste assez important ce qui est du à la fonction de backup systématique.

On peut bien faire une différence en le trafic sortant et entrant. Pendant la journée le trafic entrant est bien plus important que pendant la nuit. Ce fait prouve en quelque sort qu’il y a une fonction de backup qui envoi sur un serveur de backup toutes les données.

## Moyenne pourcentage de charge CPU

On voit que la charge de CPU est le plus grande pendant la pause de midi (12h-14h) et pendant la soirée à partir de 18h jusqu’à 2h. Cette sollicitation de la CPU est du aux étudiants qui trafiquent sur internet pendant qu’ils se trouvent chez eux dans leurs résidences.

On remarque aussi une charge de CPU à 65% ce qui est presque trop pour garantir une bonne qualité de service.

## « Daily Graph » (5 Minute average)

Sur ce graphe on peut très bien voir les horaires d’ouverture de la bibliothèque du département IF. À partir de 8 h le matin le trafic commence à augmenter et reste important jusqu’à 18h le soir et vers 20h il y en a plus. On ne remarque presque aucune activité pendant la nuit. Mme Razafinjanhary nous disait pendant la présentation que ce type de courbe est en faite la courbe parfaite car on peut n’y noter aucune incohérence. Grâce à cette courbe on pourrait facilement détecter un problème si par exemple d’un coup il y a une augmentation du trafic pendant la nuit.

## Graphe annuel (moyenne sur 1 jour)

Ce graphe présente le trafic pendant une année après installation du réseau sans fil dans un bâtiment du campus.

On sait que la rentrée scolaire est vers fin septembre donc c’est totalement normal que le trafic augmente. Par contre on s’aperçoit immédiatement qu’il y a un pic important vers le début du mois d’octobre. Ce pic ne semble pas rentrer dans le trafic habituel, probablement un ou plusieurs étudiants ont abusé avec le trafic. Grâce à ces courbes on ne peut pas conclure de quel type de téléchargement il s’agit, mais Mme Razafinjanhary nous as dit qu’il s’agissait d’un téléchargement illégal.

Début décembre il y a un autre pic très important et cette fois si il s’agissait d’une boucle sur le réseau qui à été lancé par un ordinateur. L’étudiant avait un virus sur sa machine qui provoquait ce type de trafic. Il faut donc toujours bien faire attention à ne pas installer n’importe quoi sur son ordinateur, car on est responsable de ce qui se passe avec sa propre machine.

Pendant les vacances d’hiver on détecte une forte baisse du trafic. Ce graphe permet donc de détecter des anomalies importantes sur le réseau.

## Gestion du Spam : « weekly » graph (moyenne sur 30 minutes)

Ce graphe s’étend sur une semaine et montre le nombre de messages de SPAM reçu. Il distingue les messages SPAM et non-SPAM.

Les messages envoyés pendant la journée sont plus nombreux que pendant la nuit. On s’aperçoit qu’en moyenne 42% des messages qui sont reçu correspondent à des SPAM et que le nombre augmente encore pendant le weekend. On voit que les gens travaillent moins le weekend, car le nombre de mails non-SPAM diminue pendant le weekend.