Dossier Architecture

Auteurs :
Monica Golumbeanu, Billy Pitiot, Leandro Resende Mattioli, Stefana Gartu, Soraya Belhadj Aissa, Jérôme de potter

Référence		Version	1.0
Avancement		☐ Validé	
Dernière mise à jour		☐ Validé après modif.	☐ Revalidé

Visa			
Date		Responsable	

Table des matières

1 Introduction

Le document présent constitue la réponse donnée par notre équipe de consultants à la demande de restructuration de l'architecture réseau de AIPRAO. Le dossier est structuré en deux parties. La première partie constitue un rapport décisionnel addressé à la Direction de l'AIP fournissant les informations nécessaires à sa décision de lancer le projet. La deuxième partie décrit une solution appropriée en proposant des prototypes réalisés en réponse aux différentes facettes du cahier des charges. Les annexes contiennent des éléments technologiques et organisationnels liés à l'organisation de l'environnement d'exploitation ainsi que des informations pratiques utiles pour la configuration des équipements.

2 Objectifs principaux pour la nouvelle architecture

La Direction de AIPRAO a donné un cadre définissant les objectifs pour l'évolution de l'infrastructure :

1. L'architecture doit favoriser les regroupements de moyens

   La Direction souhaite pouvoir mutualiser l'exploitation des systèmes industriels - le site central ou un autre site doit pouvoir héberger (d'une manière temporaire ou non) les ressources allouées usuellement à d'autres entités. Ainsi, une platforme industrielle virtuelle sera répartie entre des postes physiquement éloignées. L'objectif est la minimisation des transports pour les usagers.
2. L'architecture doit permettre aux entités de bénéficier de services communs pour travailler à distance.

   Des solutions de surveillance visuelle et de contrôle à distance doivent être mises en place tout en intégrant la protection des entités et de l'infrastructure globale.
3. Optimisation et performance

   La nouvelle architecture permettra de réduire les pertes de performance actuelles.
4. Mobilité

   Le déplacement d'équipement entre les sites doit être possible et facile à réaliser avec un minimum d'opérations simples de configuration.
5. Evolutivité

   L'ajout de tout matériel nouveau (serveur, platine, machine, etc.) doit être possible et facile à réaliser.
6. Securité

   La surveillance de tout le système doit être possible. Par conséquent, un système d'administration doit être mis en place.

3 Organisation de l'architecture

Les parties qui suivent contiennent la proposition d'architecture qui répond aux demandes du client. Tous les choix sont argumentés.

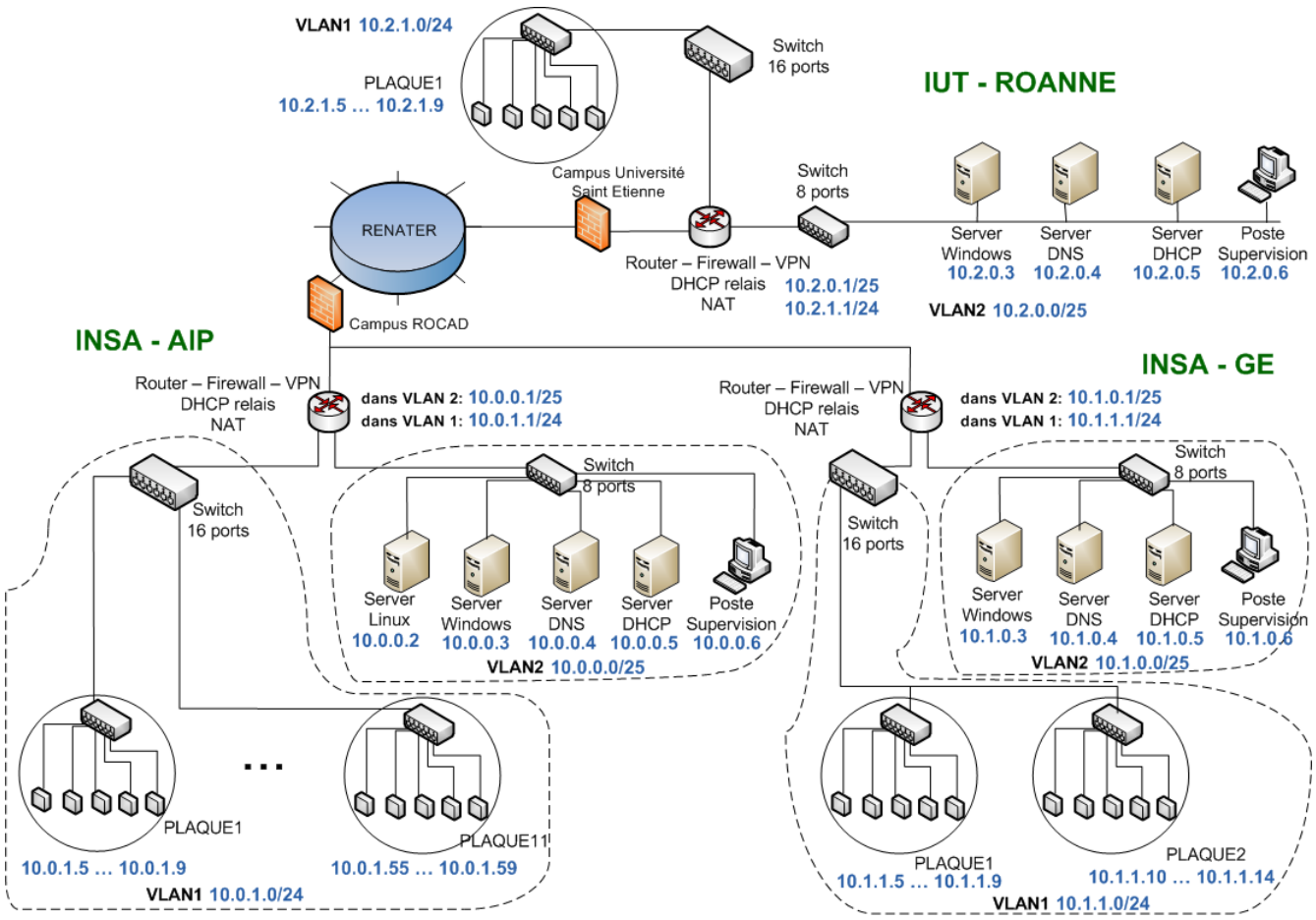


FIGURE 1 – Organisation du réseau sur trois sites

3.1 Description générale

L'architecture deployée sur les trois sites (AIP, GE, Roanne) est présentée dans la figure suivante.

La liaison entre les différents sites est réalisée par des tunnels VPN qui offrent un accès sécurisé ainsi qu'un service d'authentification efficace pour les différents utilisateurs accédant aux plateformes industrielles (voir Figure 2).

Le routeur S@N sera utilisé pour interconnecter des différentes sites. Chaque site pourra avoi plusieurs connexion VPN en parallèle. Un VPN global sera créé qui permettra le partage des ressources en toute securité.

Un utilisateur externe (étudiant, professeur, etc.) pourra se connecter à tout moment par VPN depuis l'extérieur et avoir ainsi accès aux ressources hebergées par le site auquel il s'est connecté.

Dans chaque site deux VLANs ont été mis en place. Le premier ne contient que le matériel industriel (les plaques avec les automates) et l'autre contient le reste. La configuration realisée permet de mettre en place plusieurs VLANs si on désire.

Le fait d'avoir reservé un VLAN spécialement pour l'équipement industriel permet facilement d'avoir un cloisonnement des données. En effet, le broadcast réalisé par les automates restera dans le VLAN et n'influencera pas le reste.

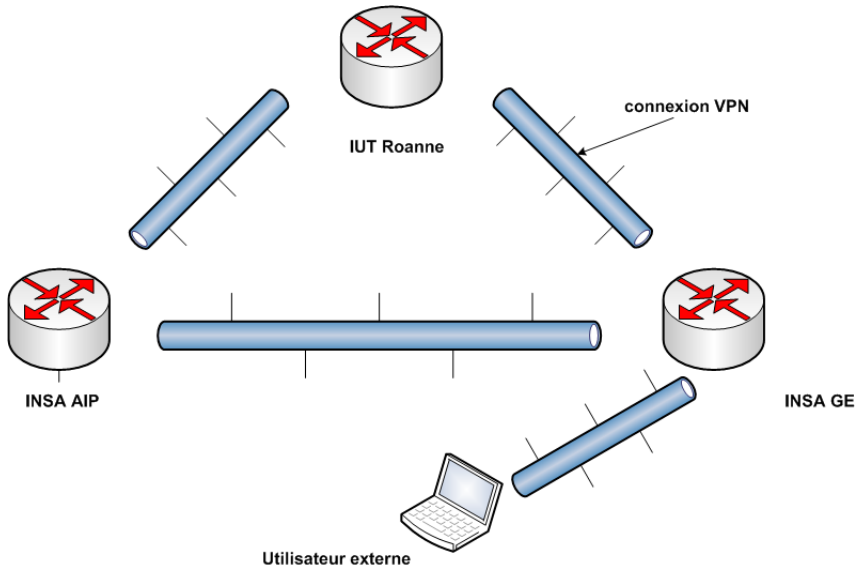


FIGURE 2 – Liaison par VPN entre les sites

Chaque site possède un serveur DHCP et un serveur DNS. Le serveur DHCP s'occupera d'associer des adresses IP en utilisant des DHCP relais entre les différents réseaux tandis que le serveur DNS gérera le nommage. Les politiques de nommage et d'addressage seront détaillées dans les sections qui suivent.

Les configurations entre les sites seront partagées par l'intermédiaire d'un système de monitoring.

3.2 Argumentation

La partie qui suit décrit comment la solution proposée peut répondre à la demande du client et atteindre les objectifs proposés.

1. L'architecture doit favoriser les regroupements de moyens

   Ceci est possible grâce à l'architecture VLAN proposée. Supposant qu'on veut mettre ensemble deux équipements physiquement éloignés, c'est possible en les mettant dans le même VLAN et donc dans le même sous réseau. Cette opération est possible via une interface web et ne nécessite pas d'actions physiques (branchement, débranchement, etc.).

2. L'architecture doit permettre aux entités de bénéficier de services communs pour travailler à distance

   Comme chaque plateforme intègre un serveur HTTP, il sera possible de se connecter de n'importe quelle poste possedant accès Internet par l'intérmediaire d'un navigateur web sur un des sites (AIP, GE, Roanne, ...). L'accès aux plaques industrielles sera donc assuré via une interface web. L'accès nécessite une authentification par VPN qui assure une connexion securisée.

3. Optimisation et performance

   Le découpage en VLANs permet le cloisonnement des données et apporte un plus des performances et d'optimisation à l'architecture. Le traffic des paquets broadcast est gardé à l'intérieur des VLANs spécialement crées ce qui allège le traffic global et augmente les performances.

4. Mobilité

Les politiques de nommage et d'adressage proposées rendent possible le déplacement et l'installation facile d'équipement entre les sites. Ces deux politiques sont détaillées dans les parties qui vont suivre.

5. Evolutivité

   L'organisation en VLANs et la politique de nommage rendent facile l'ajout de tout matériel ou toute augmentation physique du réseau.

6. Securité

   Des caméras de surveillance seront installées et pourront être suivies depuis une machine. Chaque site contient un serveur qui centralise les enregistrements vidéo. A tout moment un superviseur pourra se connecter au serveur vidéo et consulter les caméras de vidéosurveillance.

3.3 Description détaillée

Cette section comprend la description détaillée de chaque équipement utilisé dans l'architecture.

3.3.1 Routeurs Firewall VPN

Ces elements sont destinés à transporter les messages à l'intérieur des sous-réseaux. Comme une règle de base de sécurité, on utilise des firewalls pour mieux protéger ces réseaux.

De plus, le réseaux virtuel permet d'avoir un réseau logique qui intègre tous les campus et un accès distant sécurisé.

Le routeur choisi est le *VPN S@N 2000* (4 ports Ethernet 10/100Mbits, 128 tunnels VPN en parallel).

Configuration

- **Firewall**
  - bloquer toutes les requêtes venant de l'exterieur
  - redirection de port vers le S@N.
  - gerer des profils de permissions pour les postes connectés via VPN, ayant comme permission minimale l'authorisation de la connexion au serveur HTTP seulement.
- **Service DHCP** : desactivé, vu que un serveur dedié sera mis en place.
- **VPN**
  - la connexion est effectuée par l'adresse IP du routeur du campus et le port associé à cette redirection
  - 2 tunnels permanents seront établis, pour pouvoir connecter aux deux autres S@N [1]
  - permettre les connexions depuis un client VPN (rendre possible l'accès distant).

3.3.2 Switches

Les switches seront utilisés pour :
- interconnecter les élements d'une certaine plaque
- interconnecter les serveurs et postes supervision

1. Du au plan de nommage effectué voir section YY, la juxtaposition des 3 réseaux est assez simple.

– interconnecter l'ensemble des plaques
– segmenter le réseau VPN, par l'utilisation de VLANS

Les switches choisis sont :

– **TCM ESM 163F23F0 :** 16 ports, VLAN Niveau 3, 10/100 Mbits. *Utilisé pour interconnecter les plaques*
– **499 NES 181 00 :** 8 ports, 10/100 Mbits. *Utilisé à l'intérieur d'une plaque et aussi pour interconnecter les serveurs et le poste supervision.*

La division en VLANs se fera selon les critères suivants :

– un VLAN pour la manipulation.
– un VLAN pour les serveurs et le poste de supervision.

3.3.3 Serveurs DHCP, DNS et HTTP

Des serveurs DHCP et DNS dédiés seront mises en place. Leurs objectifs, sont, respectivement, l'adressage dynamique (et alors une configuration plus simple) et l'attribution de noms aux machines.

Le serveur HTTP devra fournir une interface Web pour la surveillance des plateformes et équipements industrielles. Pour raisons de sécurité, il ne sera pas accessible depuis l'exterieur sauf par connexion VPN. Chaque site aura son serveur HTTP, ce qui apporte une bon efficacité par rapport aux temps de réponse.

Pour économiser les coûts avec des logiciels, on propose d'utiliser le système d'exploitation orienté serveur *Ubuntu Server*, et le configurer avec les paquets suivants :

– Pour le serveur DHCP :
  – **isc-dhcp-server :** serveur DHCP implementé par le ICS (Internet Software Consortium)
– Pour le serveur DNS :
  – **bind9 :** le serveur BIND (Berkeley Internet Name Domain) est le serveur DNS le plus connu et utilisé, et en plus supporté par le ICS.
– Pour le serveur HTTP, les logiciels mises en place dependent des technologies de développement envisagés. Quelques logiciels associés (tous disponibles sur le système proposé) sont :
  – Apache 2
  – PHP 5
  – Ruby on Rails
  – Django
  – Glassfish
  – Mono [2]
– Pour tous les serveurs :
  – **openssh-server :** Pour permettre l'accès et la configuration à distance des serveurs.

4 Gestion du nommage et de l'adressage

Le nommage des machines est indispensable pour une lecture facile par l'œil humain et surtout pour une mémorisation et une utilisation plus intuitive qu'une adresse IP. Nous avons donc choisi une politique de nommage qui uniformisera la méthode de nommage des différentes machines sur le réseau.

2. plateforme .NET disponible sur des systèmes GNU/Linux avec support à serveurs Web ASP.NET

Le nom d'une machine sera composé de plusieurs champs distincts. Chacun de ces champs se retrouve dans l'adresse IP, ce qui permet d'associer facilement une adresse à une machine. Ces champs sont les suivants :

- Identifiant du lieu où se trouve la machine.
  Exemple : lyon_aip ==> IP correspondante : 10.0.X.X
  lyon_ge ==> IP correspondante : 10.1.X.X
  roanne_tp1 ==> IP correspondante : 10.2.X.X
- Type de l'équipement
  Exemple : Routeur ==> IP correspondante : 10.X.0.1 dans VLAN1/ 10.X.1.1 dans VLAN2
  Serveur DNS ==> IP correspondante : 10.X.0.4
  Serveur données Windows ==> IP correspondante : 10.X.0.3
  Serveur DHCP ==> IP correspondante : 10.X.0.5
  Poste Supervision ==> IP correspondante : 10.X.0.6
  Plaque_1_automate_1 ==> IP correspondante : 10.X.1.5
  Plaque_1_automate_2 ==> IP correspondante : 10.X.1.6
  Plaque_2_automate_N ==> IP correspondante : 10.X.1.2*5+(N-1) = 10.X.1.10+(N-1).

Au final nous aurons des équipements avec des noms ressemblant à celui-là : lyon_aip_plaque_1_automate_1.

L'adresse IP correspondante est : 10.0.1.5

On a choisi, dans chaque site, de créer un VLAN qui ne regroupe que l'équipement industriel (disons VLAN1) et un autre VLAN qui contient le reste du matériel (serveurs, postes, etc. - VLAN2). Cette distinction est visible au niveau des adresses IP, au niveau du troisième octet. Ainsi, les adresses des équipements industriels sont sous la forme 10.X.1.X tandis que les adresses des machines du VLAN2 son sous la forme 10.X.0.X.

L'adresse du VLAN1 sera 10.X.1.0 tandis que celle du VLAN2 sera 10.X.0.0. X diffère en fonction du site (0 pour AIP, 1 pour GE, 2 pour Roanne, etc.).

Les adresses des automates sur des plaques seront calculés d'après la formule suivante : 10.X.1.5*N°Plaque+(N°Auto
1). On modifie donc le quatrième octet. Par exemple, l'addresse de l'automate 3 situé sur la plaque 4 sur le site AIP sera 10.0.1.22.

Chaque site contient un routeur qui aura deux adresses différentes (une pour VLAN1 et une dans VLAN2). Dans VLAN1(équipement industriel) l'adresse est 10.X.1.1 et dans VLAN2 (équipement normal) est 10.X.0.1. Les routeurs utiliseront la méthode NAT pour la translation d'addresses.

La méthode mise en place permet une facilité de lecture et surtout de mise à jour en cas de déplacement d'une plaque : il suffit de mettre à jour le lieu dans lequel elle se trouve dans son nom et de changer les octets correspondants de l'adresse IP.

Pour avoir une idée globale du plan d'adressage, l'Annexe 0 fournit le plan complet, y inclut les masques des sous réseaux.

En conclusion, les adresses dans le réseau seront de la forme 10.X.Y.Z où :

- X est spécifique pour le site (0 pour AIP, 1 pour GE, 2 pour Roanne)
- Y est spécifique au type d'appareil (1 pour industriel - VLAN1, 0 pour normal - VLAN2)
- Z est spécifique à un automate sur une palque (5*N°Plaque+(N°Automate-1))

5 Système d'administration

5.1 Comparaison des solutions logicielles

Identification des critères de comparaison : Nous avons choisi une liste de critères qui serviront à déterminer notre choix entre les deux solutions logicielles Nagios couplé à MRTG et HiVison de Hirschmann et nous avons effectué un tableau comparatif entre ces deux solutions. Voici donc nos critères de comparaison :

- L'investissement initial (coût d'achat et coût d'installation)
- Coût en phase de production
- Critères techniques et performances
- Gestion des problèmes
- Statistiques
- Le confort d'utilisation et accessibilité
- Documentation et support

Critères	Nagios/MRTG	Hirschmann/HiVision
Investissement initial		
Coût d'achat	Gratuit	10 000 euros
Coût d'installation et de configuration initiale	élevé	faible
Coût en phase de production		
Coût de la licence par an	Gratuite	
Critères techniques et performances		
plugins existants	Oui	Oui
possibilité et coût d'ajout de plugins	Coût de développement	Coût d'achat
Accord en partenariat avec des entreprises	Non	Oui
Gestion des MIB	Non	Oui
Scan réseau par critères	Non	Oui
Gestion de problèmes		
Existence d'alertes	Oui	Oui
pilotage d'alertes par escalade	Non	Oui
gestion d'erreurs par niveau	Non	Oui
assistance dans la gestion des erreurs	Non	Oui
gestion d'erreurs en définissant les liens entre les différents composants réseaux	Non	Oui
Statistiques		
statistiques d'erreurs	Oui	Oui
statistiques de traffic	Oui	Oui
Confort d'utilisation et accessibilité		
monitoring à distance	Oui	Oui
monitoring par ligne téléphonique	Oui	Oui
monopolisation d'une ressource humaine 24h/24	Non	Non
Documentation et support	Faible	Bonne

Etant donné que Nagios/MRTG est une solution open source, elle présente entre autres les défauts de n'importe quelle solution similaire. La solution n'est pas très bien documentée. Elle favorise la performance à l'utilisation et la commercialisation et donc au confort et à l'accessibilité. N'ayant pas de

périmètre de fonctionnalités bien défini, nous ne pouvons pas juger de l'application de ces fonctionnalités dès le départ et il nous faudra donc une solution assez évolutive dans un temps court. Ceci n'est pas possible avec la première solution car une solution qui est non disponible doit être développée par nous étant limité en temps et en argent. 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 choisirons donc la solution payante HiVision-Hirschmann en vue de sa notoriété sur le marché.

5.2 Manuel de dépannage

5.2.1 Procédure sans intervention d'un technicien

Lorqu'une erreur ou anomalie est constatée, le personnel en charge de la surveillance du réseau doit dans un premier temps se référer au manuel de dépannage. Ce manuel permet de guider l'utilisateur en fonction de ce qu'il constate sur le terrain. Une fois l'erreur identifiée, il cherche dans le manuel les solutions proposées.

Après avoir effectué les opérations proposées, si le problème est résolu, il consigne l'erreur dans un journal ainsi que les manipulations effectuées qui ont conduit à sa résolution.

Si le problème n'est pas résolu, le manuel demande plus de détails à l'utilisateur et notamment le résultat des manipulations effectuées. Il propose alors à l'utilisateur de nouvelles manipulations plus ciblées. Et ainsi de suite jusqu'à résolution du problème.

Le manuel propose ainsi des actions de plus en plus ciblées à l'utilisateur pour résoudre le problème. Cependant, ces actions bien que ciblées sont simples et ne nécessite aucune connaissance avancée en supervision de réseau.

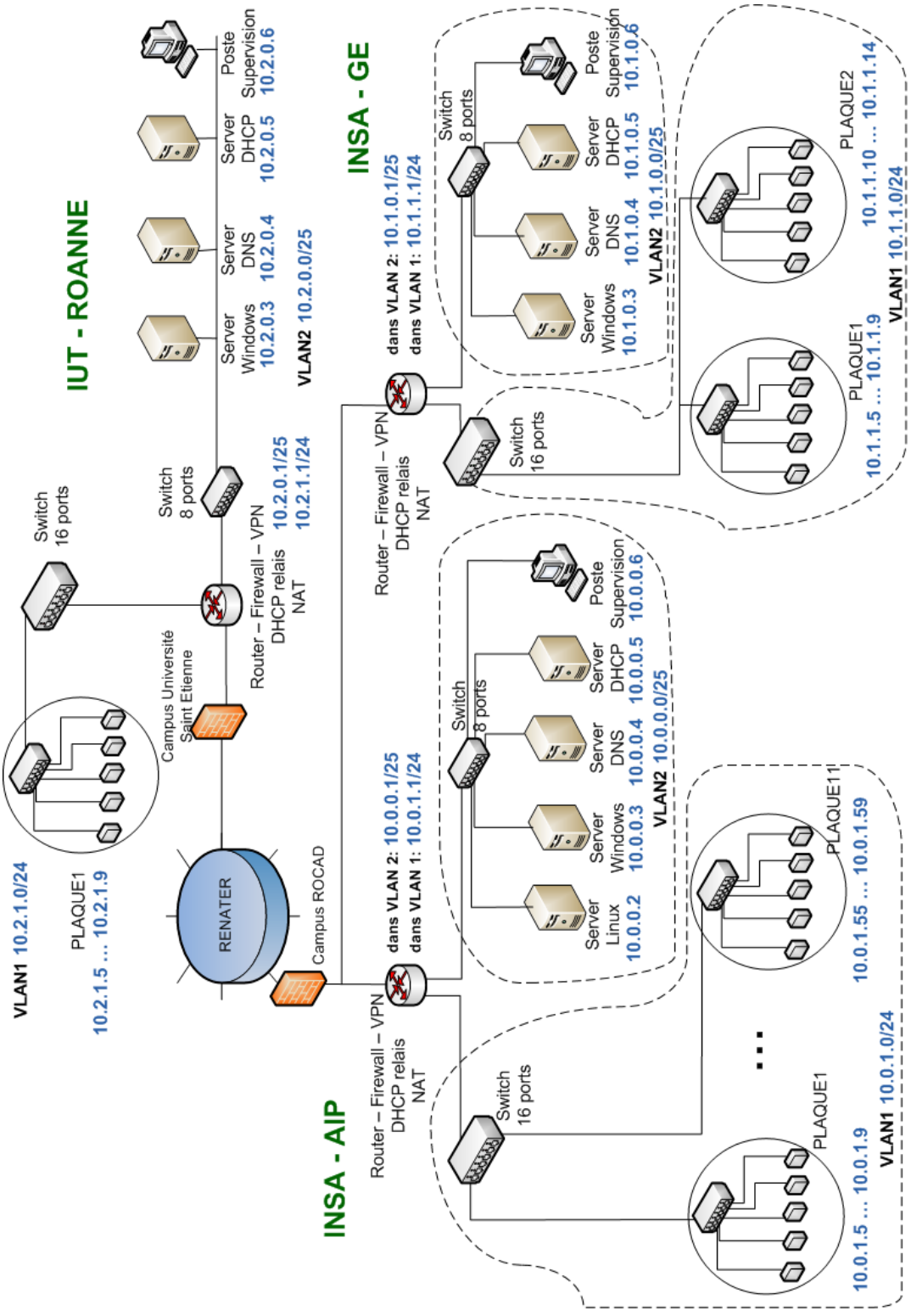
5.2.2 Procédure avec intervention d'un technicien

Si les manipulations du manuel ne suffisent pas, celui-ci invite l'utilisateur à contacter un techicien. Le manuel sauvegarde les manipulations effectuées par le superviseur et ainsi le technicien à la possibilité de les connaître. Ces informations lui permettront d'effectuer une intervention plus rapidement.

Une fois l'intervention terminée, le technicien consigne l'erreur dans le journal ainsi que les manipulations effectuées par le superviseur et par lui même.

6 Annexes

6.1 Annexe 0 - Schéma architecture



6.2 Annexe 1 - Gestion du DNS et du DHCP

6.2.1 Configurations possibles

Les paramètres obtenus pour la commande `set all` sont présentés dans la figure 3.

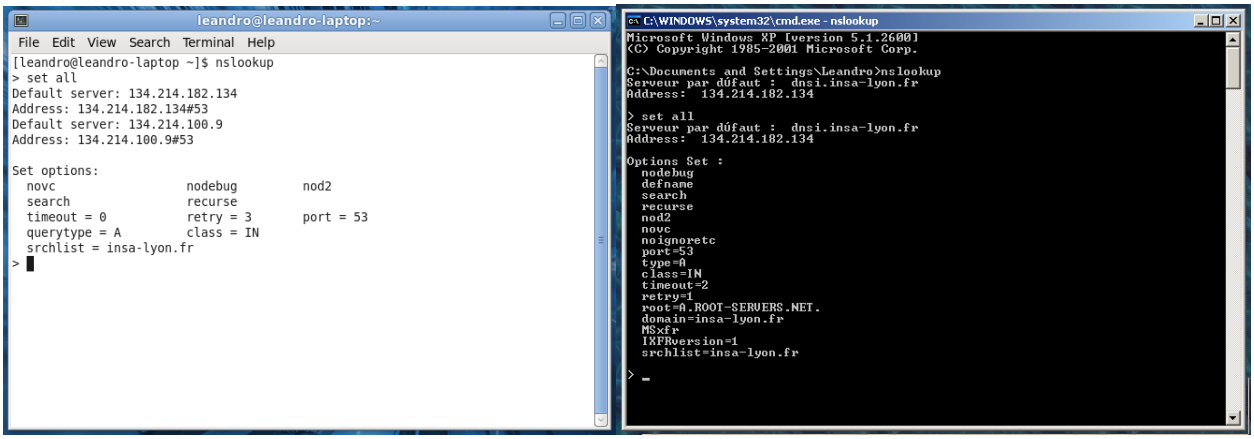


FIGURE 3 – Commande `set all` de NSLOOKUP dans les environnements WINDOWS et GNU/LINUX

Une description succinte de quelques paramètres est donnée ci-dessous :

root Serveur pour la racine du domaine (*A.ROOT-SERVERS.NET*).

domain Domaine de nom concerné (*insa-lyon.fr*).

port Port utilisé pour le serveur de domaine (53).

type Type de la requête DNS (*type A*).

retry Nombre de tentatives de la requête (*une seul tentative*).

timeout Durée d'une requête avant son expiration (*2 secondes*).

class Code de la classe de l'enregistrement DNS (*IN – Internet*).

defname Ajouter le nom du domaine par défaut à des requêtes simples (qui ne contiennent pas de période).

recurse Demander à autres serveurs si on n'a pas l'information.

srchlist Liste des domaines utilisés par le paramètre *search*. Le domaine défini par une commande `set domain` sera remplacé par le premier item de la liste. (*insa-lyon.fr*).

search Cette option ajoute les domaines stockés par le paramètre *srchlist* à la requête jusqu'à réception d'une réponse, sauf si la requête est finie par une période (.).

novc Ne pas utiliser un circuit virtuel.

nodebug Sans information de debogage.

6.2.2 Résolution du nom servif-baie

La figure 4 montre le résultat de la commande.

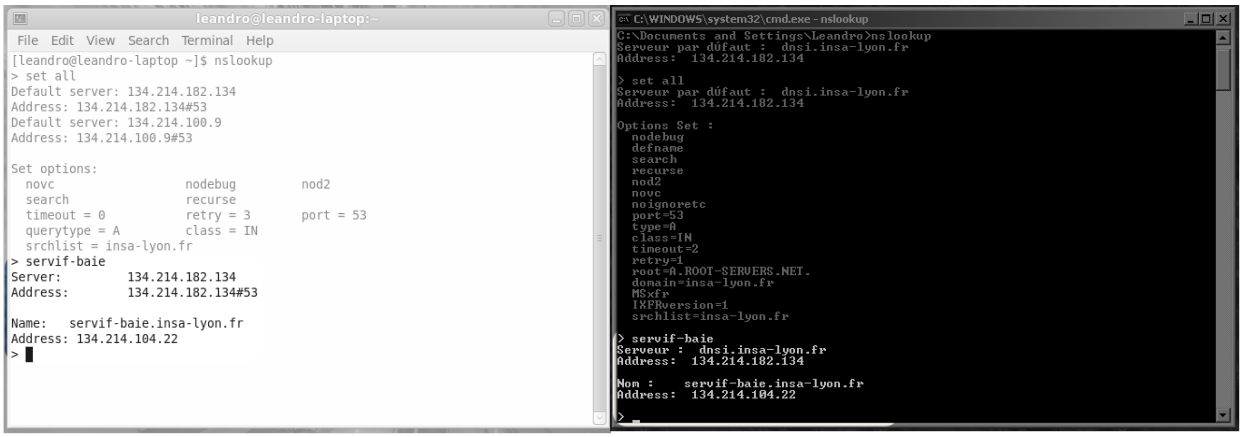


FIGURE 4 – Résolution de la machine SERVIF-BAIE

Il n'y a pas besoin du nom complet (avec le nom du domaine inclus) à cause des paramètres DEFNAME et DOMAIN, vus dans la section précedent. Avec un nom de domaine défini et la option defname activé, les requêtes d'un seul element (requêtes sans periode) sont automatiquement completées. L'adresse de SERVIF-BAIE est 134.214.104.22 et le type d'enregistrement utilisé est le type A (un IPv4, 4 octets).

6.2.3 Résolution du mail

La résolution de la machine MAIL est montré dans la figure 5. Son nom est MAIL.INSA-LYON.FR (alias pour DSI04.INSA-LYON.FR).

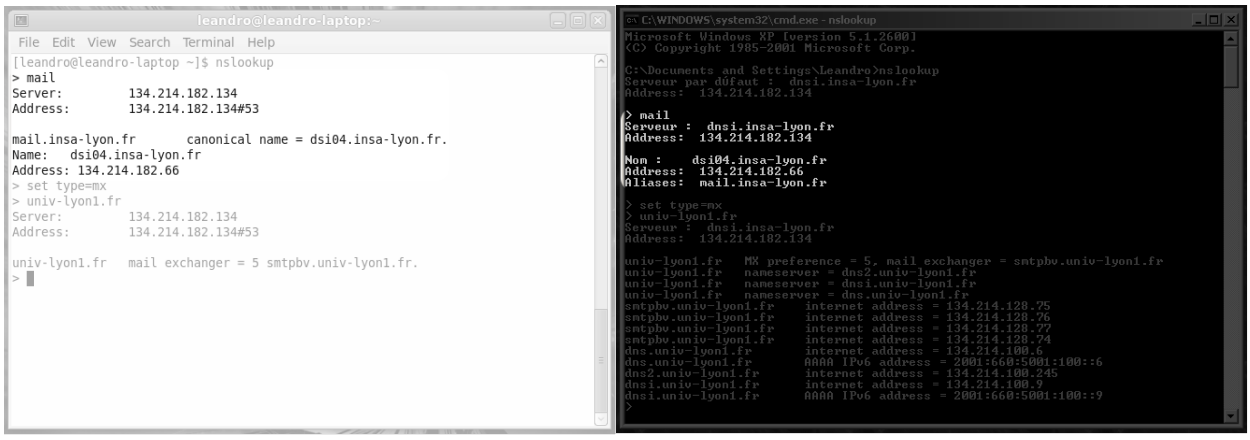


FIGURE 5 – Résolution de MAIL

Pour obtenir l'adresse du serveur de mail de LYON 1, on peut changer le type de la requête DNS, avec la commande `set type=mx` (figure 6). Avec cette commande, on peut trouver les serveur MAIL liés à un certain domaine. Le serveur de mail de Lyon 1 est, donc, SMTPBV.UNIV-LYYON1.FR.

Vu qu'il existe seulement un serveur, la valeur de la préference (dans ce cas, 5) n'est pas utile. Par contre, on peut voir qu'il existe une répartition de charge entre 4 machines (134.214.128.75 ... 134.214.128.77).

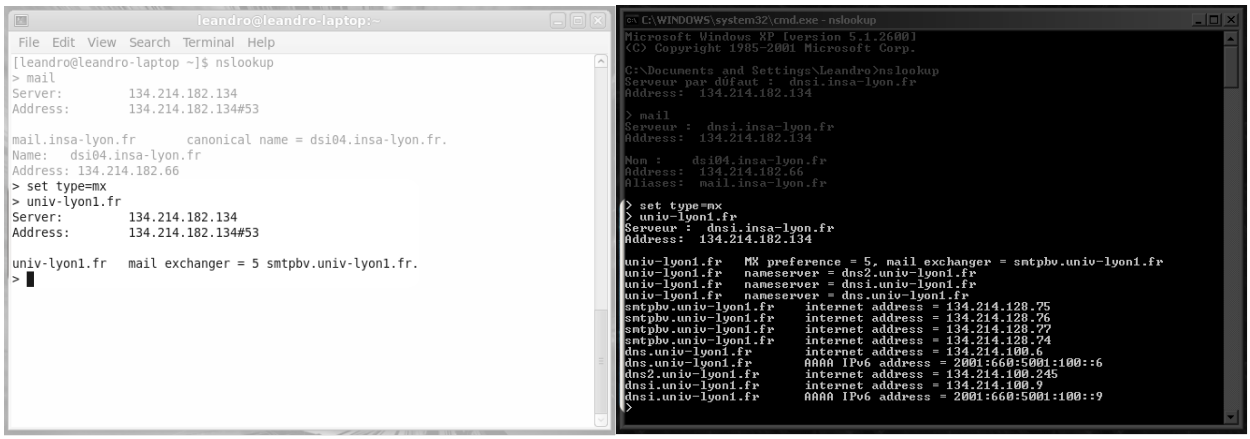


FIGURE 6 – Résolution de UNIV-LYON1

6.2.4 Obtention du nom de 134.214.61.235

Comme montré dans la figure 7, la machine 134.214.61.235 a comme nom IF-4207.INSA-LYON.FR et le type d'enregistrement utilisé est le type A.

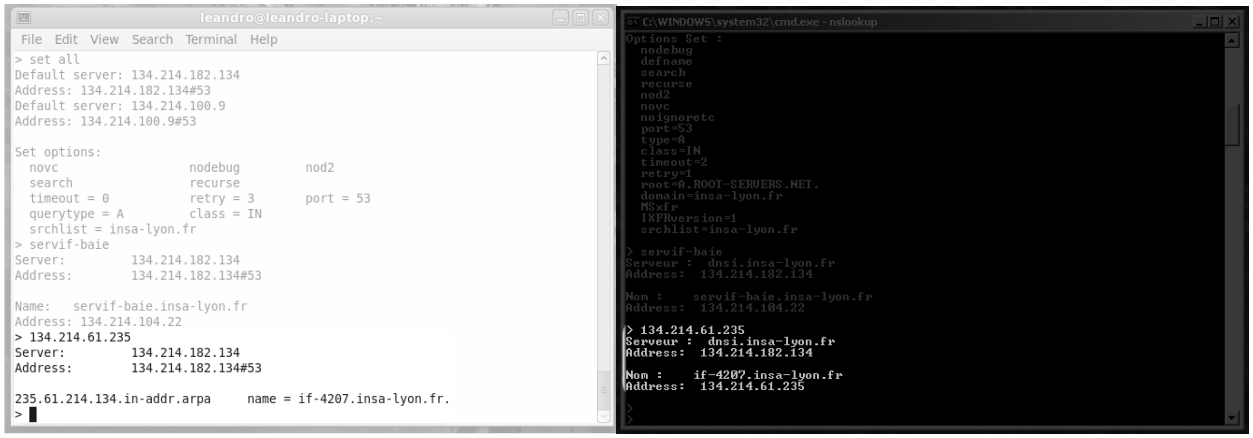


FIGURE 7 – Obtention du nom de 134.214.61.135

6.2.5 Résolution du host www.google.fr

Finalement, pour le nom WWW.GOOGLE.FR (figure 8), on peut voir que les adresses WWW.GOOGLE.FR et WWW.GOOGLE.COM sont redirigées vers WWW.L.GOOGLE.COM. Pour bien pouvoir répondre à toutes requêtes, Google utilise des plusieurs serveurs pour diviser la charge. Le serveur est du type recursif et donc utilise un cache pour répondre les requêtes. Ceci est pas forcement à jour. On parle alors d'une *réponse ne faisant pas autorité*.

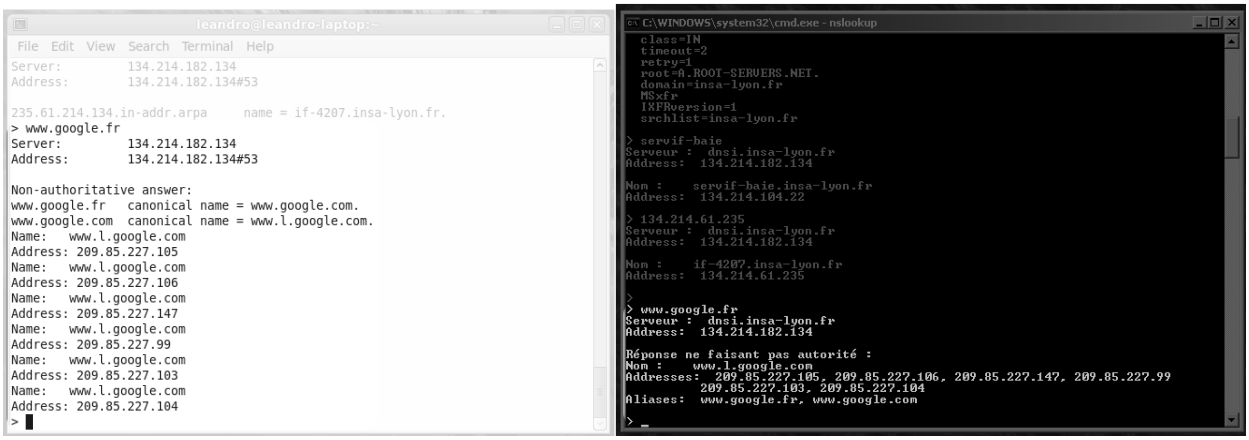


FIGURE 8 – Résolution du nom WWW.GOOGLE.FR

6.2.6 Conclusion

À partir des informations que nous venons de faire, on peut constater que les types d'enregistrements concernés sont :

A : Du à l'obtention de l'adresse IPv4

CNAME : Utilisation des *alias*

SOA : Informations d'authorité de la zone DNS (serveur primaire, etc).

6.3 Annexe 2 - Proof of concept de la solution d'interconnexion

6.3.1 VLAN Tagged/Untagged

Il est important de comprendre l'usage des VLANs "tagged" et "untagged". Un port peut être un membre "untagged" si il ne fait partie que d'un VLAN. Par contre, dans le cas où un port peut correspondre à plusieurs VLANs, il faut qu'il soit "tagged" pour chacun des VLANs auxquels il appartient.

Typiquement, les station finales seront marquées comme membres "untagged" d'une VLAN. De l'autre côté, les connexions entre switchs devraient être "tagged". Dû à cette configuration, il sera possible de permettre aux stations finales d'une même VLAN de partager les connexions avec autres VLANs.

Les adresses des switches utilisées son 134.214.105.222 pour le switch 2 et 134.214.105.223 pour le switch 3. En utilisant le modèle 3COM-4400 et l'application SUPER STACK pour gérer et maintenir les switches, les menus de configuration des switches se trouvent dans Bridge → VLAN → Display/Edit.

6.3.2 Configuration du switch 2

En tenant compte de la configuration des switches, nous pouvons observer que le VLAN 4 est connecté au switch 1 et son port 24 est connectée au switch 2. Ainsi, pour que le switch 2 puisse y accéder, il faut qu'on marque le port 1 du switch 2 comme "tagged", comme la figure nous montre ci-dessous.

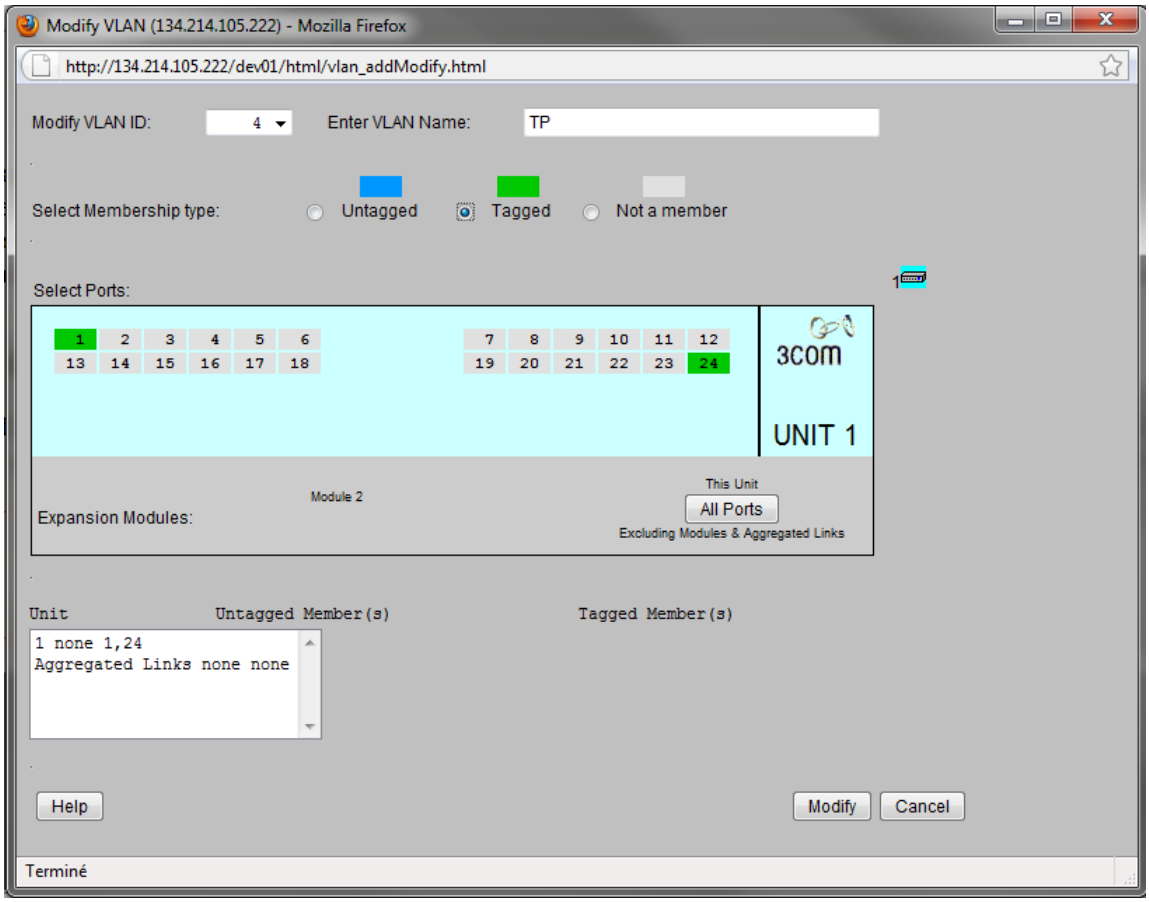


Figure 9: Configuration du switch 2

On observe aussi que le port 24 est marqué comme “tagged”. C’est à cause du lien avec le switch 3. Ce lien permet au switch 3 d’accéder au VLAN 4.

6.3.3 Configuration du switch 3

Pour la configuration du switch 3, nous avons utilisé presque la même configuration que pour le switch 2. La différence c’est que nous avons marqué le port 6 comme “untagged” pour qu’elle soit accessible à partir d’un ordinateur du VLAN 4. La figure ci-dessous nous montre sa configuration.

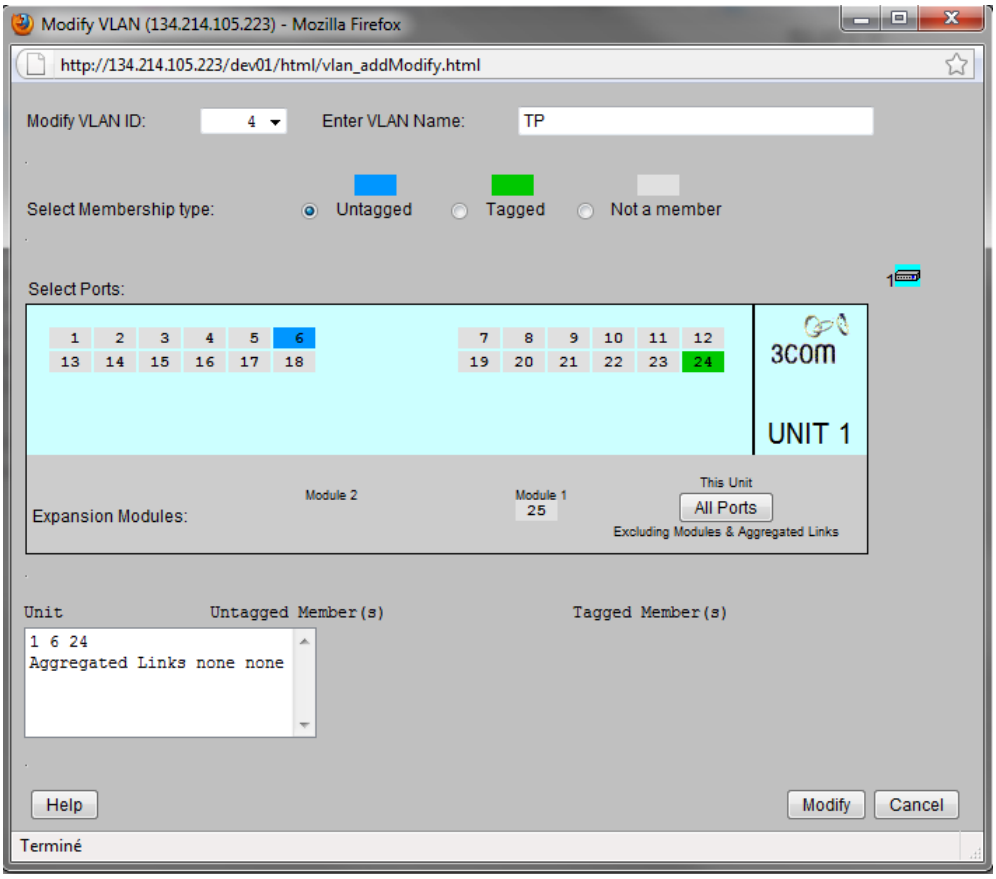


Figure 10: Configuration du switch 3

Pour les autres ports, nous les avons marquées comme Not a member pour qu’ils ne soient pas accessibles sur le VLAN4 - TP.

6.3.4 Conclusion

En utilisant cette configuration, nous nous assurons qu’on peut connecter un PC au port 6 du switch 3 qui peut accéder au VLAN4 - TP et que les ports 1-5, 7-23 ne peuvent pas y accéder.

6.4 Annexe 3 - Monitoring réseau

6.4.1 Ipconfig Machine Windows

Machine Windows : 134.214.105.165

Masque de sous réseau : 255.255.255.0

Passerelle par défaut : 134.214.105.1

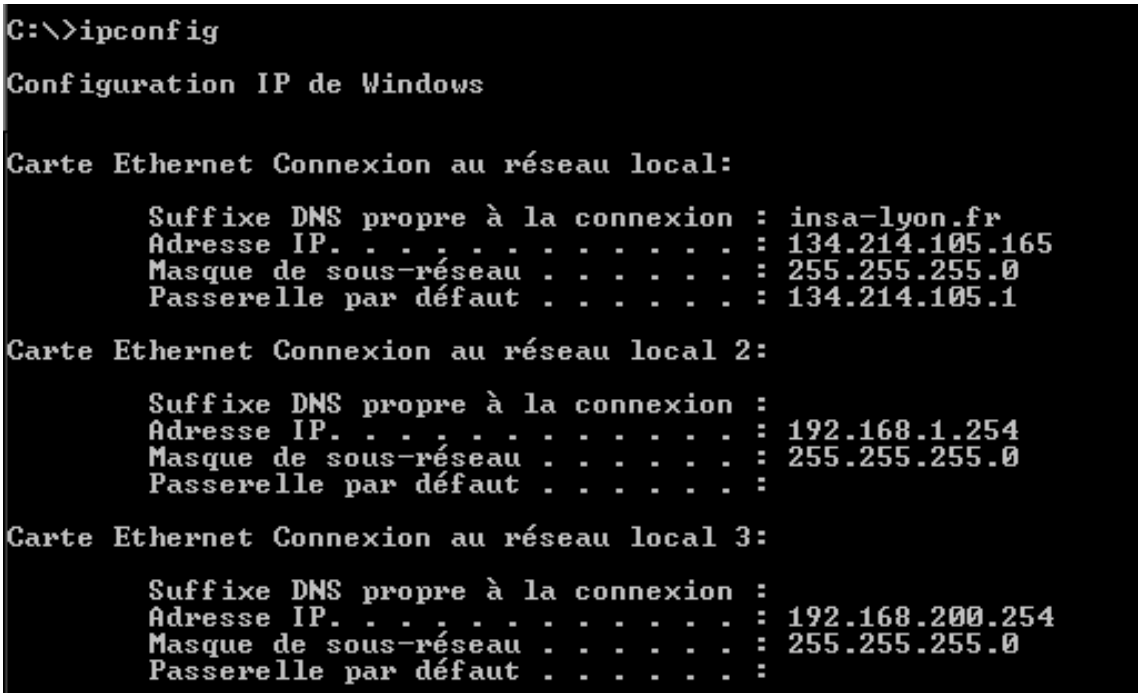


FIGURE 11 – ipconfig Machine Windows

6.4.2 Ipconfig Serveur Nagios

Serveur Nagios : 134.214.105.156

Masque de sous réseau : 255.255.255.0

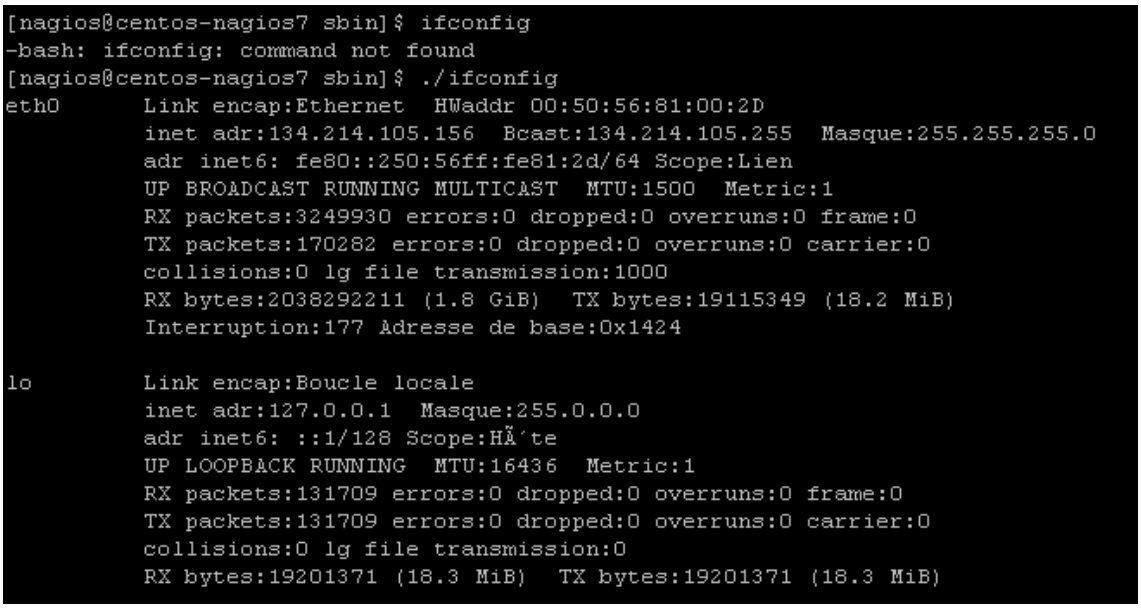


FIGURE 12 – ifconfig Serveur Nagios

On constate que les deux machines sont sur le même sous-réseau.

6.4.3 Netstat Machine Windows

Netstat

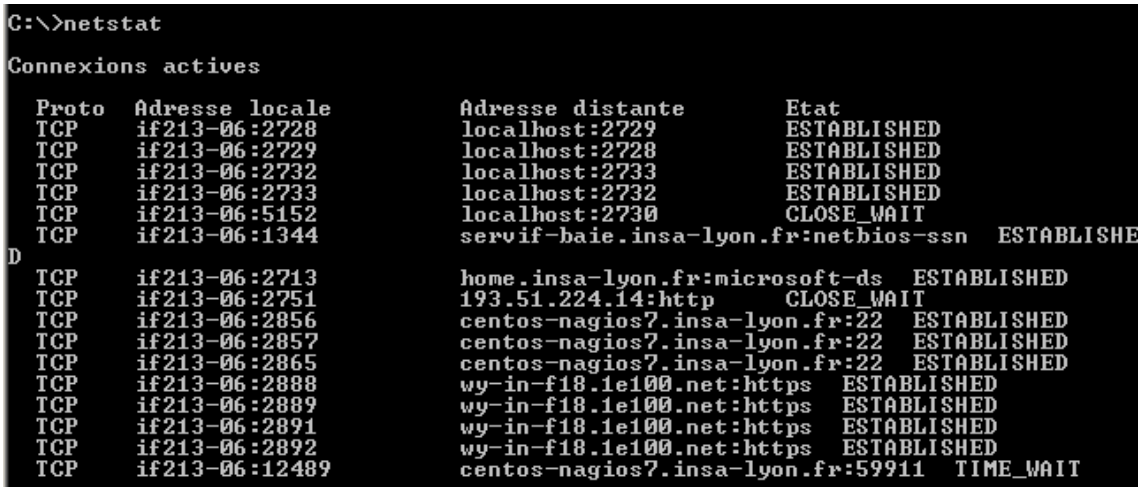


FIGURE 13 – netstat

A chaque ligne correspond une connexion réseau établie entre la machine windows locale et une machine distante.

Les colonnes sont les suivantes :

Proto : Indique le protocole de communication utilisé. Celui-ci peut être : TCP, UDP, TCPv6 et UDPv6.

Adresse locale : Indique le nom ou l'adresse IP de la machine sur laquelle netstat est exécutée ainsi que le port de sortie de la connexion.

Adresse distante : Indique le nom ou l'adresse IP de la machine distante avec laquelle la connexion est établie ainsi que le port d'entrée de la connexion.

Etat : Indique l'état de la connexion. Celui-ci peut être :

- LISTENING -> En écoute
- ESTABLISHED -> Etablie
- TIME_WAIT -> En attente
- CLOSE_WAIT -> En attente de fermeture

Ici, on peut voir différentes connexions :

- Les quatres premières connexion sont établies entre if213-06 et la machine locale
- La cinquième est en attente de fermeture entre if213-06 et la machine locale
- Les sixième et septième connexions sont établies avec insa-lyon.fr pour l'accès à servif-baie et home
- Trois connexions sont établies et une est en attente avec le serveur nagios
- Les autres lignes correspondent à des connexions à des machines distantes en utilisant les protocoles http et https.

Netstat -a

```
Connexions actives

  Proto  Adresse locale         Adresse distante       Etat
  TCP    if213-06:epmap         if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:microsoft-ds  if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:912           if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:1040          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:sapgw89       if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:5555          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:5666          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:7777          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:12489         if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:1043          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:2725          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:2728          localhost:2729         ESTABLISHED
  TCP    if213-06:2729          localhost:2728         ESTABLISHED
  TCP    if213-06:2732          localhost:2733         ESTABLISHED
  TCP    if213-06:2733          localhost:2732         ESTABLISHED
  TCP    if213-06:5152          if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:5152          localhost:2730         CLOSE_WAIT
  TCP    if213-06:netbios-ssn   if213-06.insa-lyon.fr:0  LISTENING
  TCP    if213-06:1344          servif-baie.insa-lyon.fr:netbios-ssn  ESTABLISHE
D
  TCP    if213-06:2713          home.insa-lyon.fr:microsoft-ds  ESTABLISHED
  TCP    if213-06:2751          193.51.224.14:http     CLOSE_WAIT
  TCP    if213-06:2856          centos-nagios7.insa-lyon.fr:22  ESTABLISHED
  TCP    if213-06:2857          centos-nagios7.insa-lyon.fr:22  ESTABLISHED
  TCP    if213-06:2865          centos-nagios7.insa-lyon.fr:22  ESTABLISHED
  TCP    if213-06:2918          centos-nagios7.insa-lyon.fr:22  ESTABLISHED
  TCP    if213-06:2928          wy-in-f19.1e100.net:https  TIME_WAIT
  TCP    if213-06:2956          wy-in-f18.1e100.net:https  ESTABLISHED
  TCP    if213-06:2959          wy-in-f18.1e100.net:https  ESTABLISHED
  TCP    if213-06:12489         centos-nagios7.insa-lyon.fr:33052  TIME_WAIT
  UDP    if213-06:microsoft-ds  *:*
  UDP    if213-06:isakmp        *:*
  UDP    if213-06:1346          *:*
  UDP    if213-06:ms-sql-m      *:*
  UDP    if213-06:4500          *:*
  UDP    if213-06:5555          *:*
  UDP    if213-06:ntp           *:*
  UDP    if213-06:1025          *:*
  UDP    if213-06:1039          *:*
  UDP    if213-06:1041          *:*
  UDP    if213-06:1047          *:*
  UDP    if213-06:1053          *:*
  UDP    if213-06:1054          *:*
  UDP    if213-06:1056          *:*
  UDP    if213-06:1093          *:*
  UDP    if213-06:ntp           *:*
  UDP    if213-06:netbios-ns    *:*
  UDP    if213-06:netbios-dgm   *:*
  UDP    if213-06:ntp           *:*
  UDP    if213-06:ntp           *:*
```

FIGURE 14 – netstat -a

La commande netstat -a présente en plus des connexions de netstat, les connexions qui sont en écoute. Les connexions dont l'adresse distante est * :* sont en écoute sur le port de l'adresse locale.

Netstat -e

```
C:\>netstat -e
Statistiques de l'interface

                           Reçus         Emis

Octets                   279628894        99864748
Paquets unicast             309062          311728
Paquets non monodiffusion                36235              148
Rejets                          0               0
Erreurs                         0               0
Protocoles inconnus         56744
```

FIGURE 15 – netstat -e

La commande netstat -e présente les informations concernant l'ensemble des connexions. On obtient ainsi le nombre d'octets, de paquets unicast et de paquets non monodiffusion reçus et émis. On peut également connaître le nombre de rejets ou d'erreurs.

Netstat -r

```
C:\>netstat -r

Table de routage
===========================================================================
Liste d'Interfaces
0x1 ........................... MS TCP Loopback interface
0x2 ...00 1f d0 a7 6e 33 ...... Realtek RTL8168C(P)/8111C(P) PCI-E Gigabit Ether
net NIC - Teefer2 Miniport
0x10005 ...00 00 00 00 00 fe ...... WindRiver ULIP - Teefer2 Miniport
0x20003 ...7a 7a c0 a8 c8 fe ...... WindRiver WRTAP - Teefer2 Miniport
===========================================================================
===========================================================================
Itinéraires actifs :
Destination réseau    Masque réseau  Adr. passerelle   Adr. interface Métrique
          0.0.0.0          0.0.0.0    134.214.105.1  134.214.105.165      10
        127.0.0.0        255.0.0.0        127.0.0.1        127.0.0.1       1
    134.214.105.0    255.255.255.0  134.214.105.165  134.214.105.165      10
  134.214.105.165  255.255.255.255        127.0.0.1        127.0.0.1      10
  134.214.255.255  255.255.255.255  134.214.105.165  134.214.105.165      10
      192.168.1.0    255.255.255.0    192.168.1.254    192.168.1.254      30
    192.168.1.254  255.255.255.255        127.0.0.1        127.0.0.1      30
    192.168.1.255  255.255.255.255    192.168.1.254    192.168.1.254      30
    192.168.200.0    255.255.255.0  192.168.200.254  192.168.200.254      30
  192.168.200.254  255.255.255.255        127.0.0.1        127.0.0.1      30
  192.168.200.255  255.255.255.255  192.168.200.254  192.168.200.254      30
        224.0.0.0        240.0.0.0  134.214.105.165  134.214.105.165      10
        224.0.0.0        240.0.0.0    192.168.1.254    192.168.1.254      30
        224.0.0.0        240.0.0.0  192.168.200.254  192.168.200.254      30
  255.255.255.255  255.255.255.255  134.214.105.165  134.214.105.165      1
  255.255.255.255  255.255.255.255    192.168.1.254    192.168.1.254      1
  255.255.255.255  255.255.255.255  192.168.200.254  192.168.200.254      1
Passerelle par défaut :     134.214.105.1
===========================================================================
Itinéraires persistants :
  Aucun
```

FIGURE 16 – netstat -r

La commande netstat -r présente la table de routage ainsi que les itinéraires actifs. Pour chaque destination, est renseigné, son masque de sous réseau, l'adresse de sa passerelle, l'adresse de son interface et sa métrique. Certaines adresses sont remarquables :

127.0.0.0 -> Adresse de Loopback

134.214.105.165 -> Adresse de la machine Windows

134.214.255.255 -> Adresse de broadcast

192.168.0.1 -> Adresse de réseau privé

Netstat -n

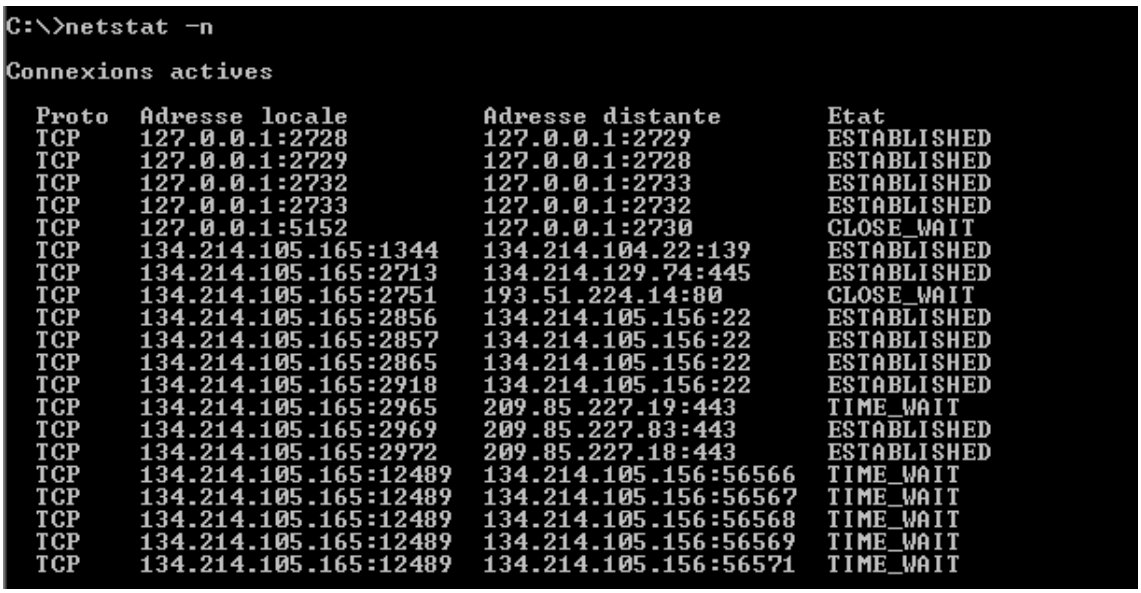


FIGURE 17 – netstat -n

La commande netstat -n présente les connexions actives. A la différence de netstat, cette commande ne présente jamais les noms de la machine locale et des machines distantes mais uniquement les adresse IP. Les connexions LISTENING ne sont pas présente dans cette liste puisque n'étant pas actives.

Netstat -p TCP

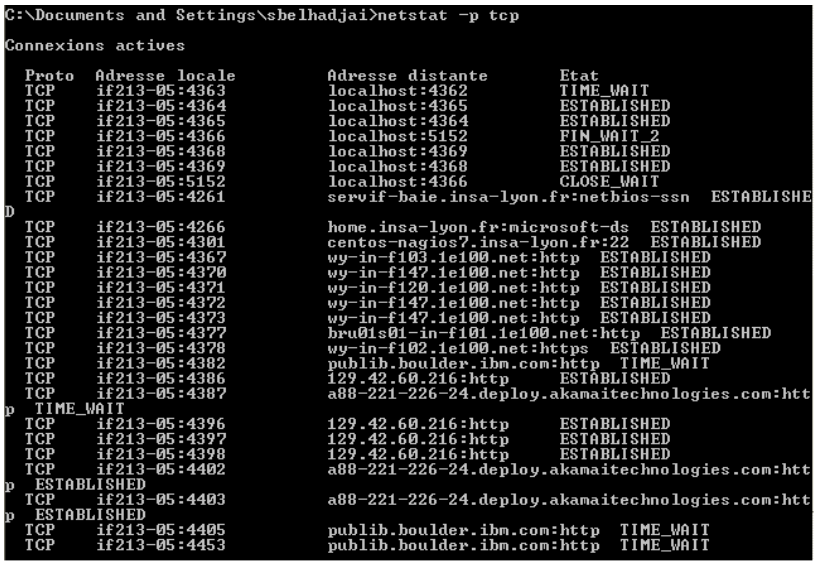


FIGURE 18 – netstat -p tcp

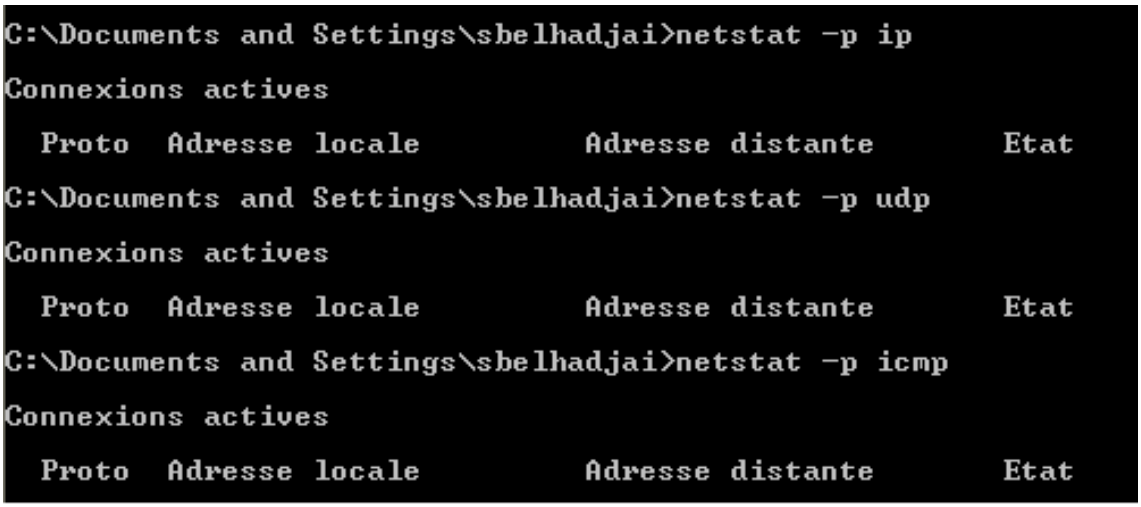


FIGURE 19 – netstat -p udp , -p ip, -p icmp

La commande netstat -p présente les connexions active pour le protocole précisé. On remarque ici qu'il n'y a que des connexions TCP.

Netstat -s

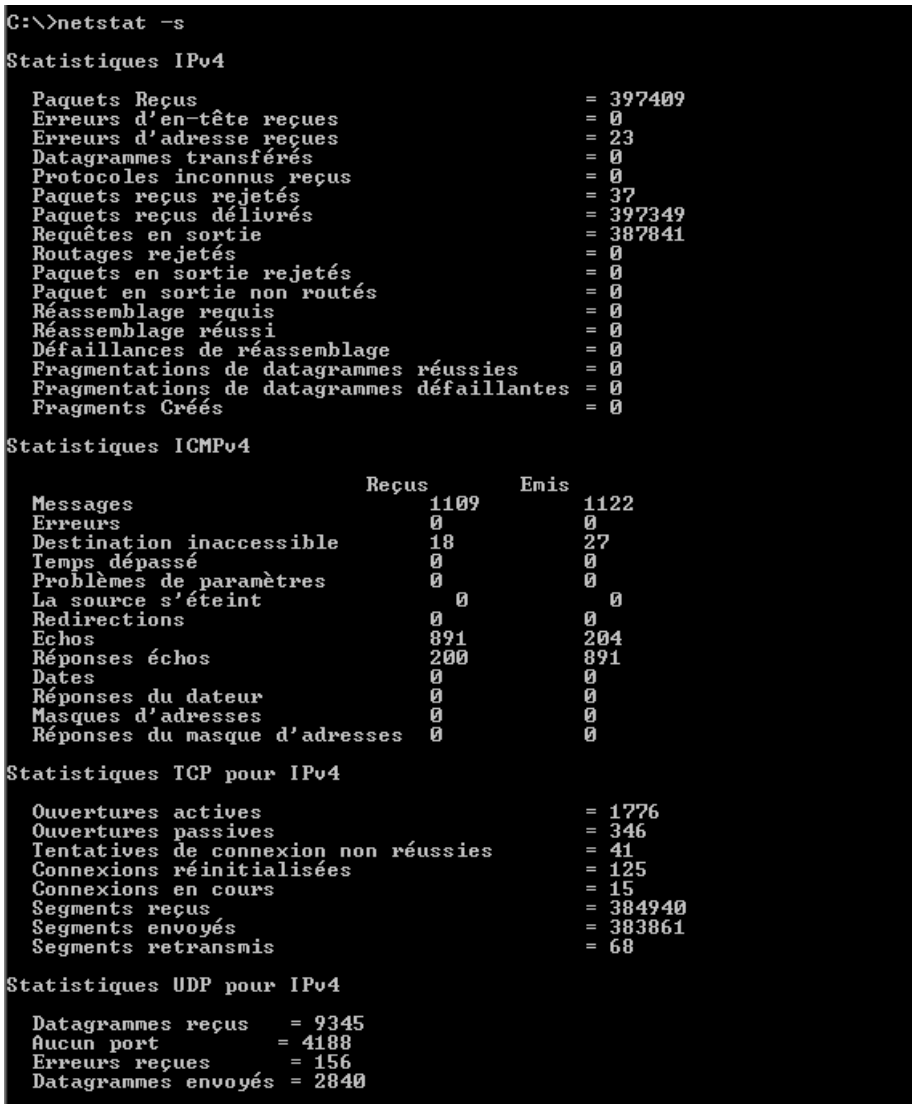


FIGURE 20 – netstat -s

La commande netstat -s pésente les statistiques de toutes les connexions par protocole.

6.4.4 Netstat Serveur nagios

Netstat

On éxécute les mêmes commandes sur le serveur nagios.

FIGURE 21 – netstat

Netstat -a

FIGURE 22 – netstat -a

Netstat -e

FIGURE 23 – netstat -e

Netstat -r

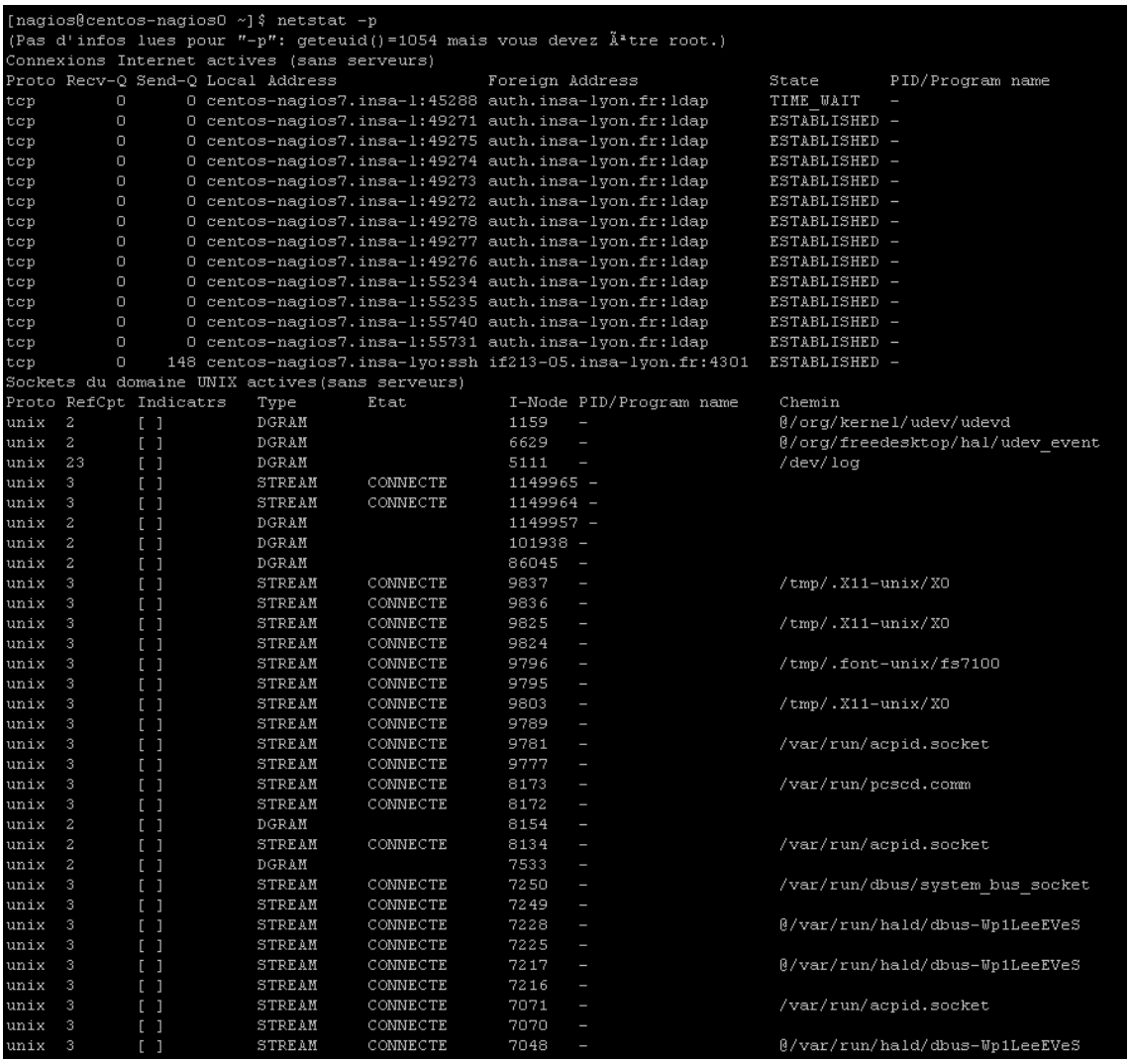
FIGURE 24 – netstat -r

Netstat -n

```
[nagios@centos-nagios0 ~]$ netstat -n
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 134.214.105.156:49271       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49275       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49274       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49273       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49272       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49278       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49277       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:49276       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:55234       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:55235       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:55740       134.214.182.100:389         ESTABLISHED
tcp        0      0 134.214.105.156:55731       134.214.182.100:389         ESTABLISHED
tcp        0      0 ::ffff:134.214.105.156:22   ::ffff:134.214.105.164:4301 ESTABLISHED
Sockets du domaine UNIX actives(sans serveurs)
Proto RefCpt Indicatrs   Type       Etat          I-Node Chemin
unix  2      [ ]         DGRAM                    1159   @/org/kernel/udev/udevd
unix  2      [ ]         DGRAM                    6629   @/org/freedesktop/hal/udev_event
unix  23     [ ]         DGRAM                    5111   /dev/log
unix  3      [ ]         STREAM     CONNECTE      1149965
unix  3      [ ]         STREAM     CONNECTE      1149964
unix  2      [ ]         DGRAM                    1149957
unix  2      [ ]         DGRAM                    101938
unix  2      [ ]         DGRAM                    86045
unix  3      [ ]         STREAM     CONNECTE      9837   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTE      9836
unix  3      [ ]         STREAM     CONNECTE      9825   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTE      9824
unix  3      [ ]         STREAM     CONNECTE      9796   /tmp/.font-unix/fs7100
unix  3      [ ]         STREAM     CONNECTE      9795
unix  3      [ ]         STREAM     CONNECTE      9803   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTE      9789
unix  3      [ ]         STREAM     CONNECTE      9781   /var/run/acpid.socket
unix  3      [ ]         STREAM     CONNECTE      9777
unix  3      [ ]         STREAM     CONNECTE      8173   /var/run/pcscd.comm
unix  3      [ ]         STREAM     CONNECTE      8172
unix  2      [ ]         DGRAM                    8154
unix  2      [ ]         STREAM     CONNECTE      8134   /var/run/acpid.socket
unix  2      [ ]         DGRAM                    7533
unix  3      [ ]         STREAM     CONNECTE      7250   /var/run/dbus/system_bus_socket
```

FIGURE 25 – netstat -n

Netstat -p



```
[nagios@centos-nagios0 ~]$ netstat -p
(Pas d'infos lues pour "-p": geteuid()=1054 mais vous devez Ã*tre root.)
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Local Address               Foreign Address             State       PID/Program name
tcp        0      0 centos-nagios7.insa-l:45288 auth.insa-lyon.fr:ldap      TIME_WAIT   -
tcp        0      0 centos-nagios7.insa-l:49271 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49275 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49274 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49273 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49272 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49278 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49277 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:49276 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:55234 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:55235 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:55740 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0      0 centos-nagios7.insa-l:55731 auth.insa-lyon.fr:ldap      ESTABLISHED -
tcp        0    148 centos-nagios7.insa-lyo:ssh if213-05.insa-lyon.fr:4301  ESTABLISHED -
Sockets du domaine UNIX actives(sans serveurs)
Proto RefCpt Indicatrs   Type       Etat          I-Node PID/Program name    Chemin
unix  2      [ ]         DGRAM                    1159   -                   @/org/kernel/udev/udevd
unix  2      [ ]         DGRAM                    6629   -                   @/org/freedesktop/hal/udev_event
unix  23     [ ]         DGRAM                    5111   -                   /dev/log
unix  3      [ ]         STREAM     CONNECTE      1149965 -
unix  3      [ ]         STREAM     CONNECTE      1149964 -
unix  2      [ ]         DGRAM                    1149957 -
unix  2      [ ]         DGRAM                    101938 -
unix  2      [ ]         DGRAM                    86045  -
unix  3      [ ]         STREAM     CONNECTE      9837   -                   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTE      9836   -
unix  3      [ ]         STREAM     CONNECTE      9825   -                   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTE      9824   -
unix  3      [ ]         STREAM     CONNECTE      9796   -                   /tmp/.font-unix/fs7100
unix  3      [ ]         STREAM     CONNECTE      9795   -
unix  3      [ ]         STREAM     CONNECTE      9803   -                   /tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTE      9789   -
unix  3      [ ]         STREAM     CONNECTE      9781   -                   /var/run/acpid.socket
unix  3      [ ]         STREAM     CONNECTE      9777   -
unix  3      [ ]         STREAM     CONNECTE      8173   -                   /var/run/pcscd.comm
unix  3      [ ]         STREAM     CONNECTE      8172   -
unix  2      [ ]         DGRAM                    8154   -
unix  2      [ ]         STREAM     CONNECTE      8134   -                   /var/run/acpid.socket
unix  2      [ ]         DGRAM                    7533   -
unix  3      [ ]         STREAM     CONNECTE      7250   -                   /var/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTE      7249   -
unix  3      [ ]         STREAM     CONNECTE      7228   -                   @/var/run/hald/dbus-Wp1LeeEVeS
unix  3      [ ]         STREAM     CONNECTE      7225   -
unix  3      [ ]         STREAM     CONNECTE      7217   -                   @/var/run/hald/dbus-Wp1LeeEVeS
unix  3      [ ]         STREAM     CONNECTE      7216   -
unix  3      [ ]         STREAM     CONNECTE      7071   -                   /var/run/acpid.socket
unix  3      [ ]         STREAM     CONNECTE      7070   -
unix  3      [ ]         STREAM     CONNECTE      7048   -                   @/var/run/hald/dbus-Wp1LeeEVeS
```

FIGURE 26 – netstat -p

Netstat -s

FIGURE 27 – netstat -s

6.4.5 Ping

La commande ping permet de savoir si une machine ou un serveur est accessible depuis une autre (machine sur laquelle est exécuter la commande). Cette commande envoie un "ping" à la machine distante et attend une réponse. Si la machine n'est pas accessible, la commande se terminera pas un échec suite à un timeout. L'exécution de la commande ping affiche le temps de réponse de la machine distante. Si la commande affiche un temps de réponse très long, c'est qu'il est possible qu'il y ai un problème sur le réseau.

Dans notre cas, le message suivant s'affiche sur la console :

Réponse de 134.214.105.221 : octets=32 temps=1 ms TTL=254

Dans ce cas, la machine distante à répondu et la réponse a été rapide. Il n'y a donc a priori pas de problème sur le réseau.

6.4.6 Sniffer de paquet : Wireshark

Le logiciel Wireshark permet d'analyser les trames circulant sur un réseau. L'analyse de la requête ping permet de connaitre la composition de la trame. Celle-ci fait 74 octets est composé d'un entête ethernet, d'un entête ip, d'un entête icmp et de données. L'entête ethernet présente les adresses MAC de source et de destination de la trame et indique que le protocole ip est utilisé pour la couche supérieure. L'entête ip présente la version (ipv4), sa taille (20 octets), l'utilisation du protocole de controle ICMP et checksum et les adresses ip de source et de destination du paquet. L'entête ICMP contient des contrôles de validité du paquet. Les données sont écrites sur 32 octets.

6.4.7 MIB

MIB est une base de données regroupant les informations concernant le métériel sur lequel elle est installée. Ces informations sont utiles pour gérer un réseau. L'accès à la MIB se fait à l'aide du protocole snmp.

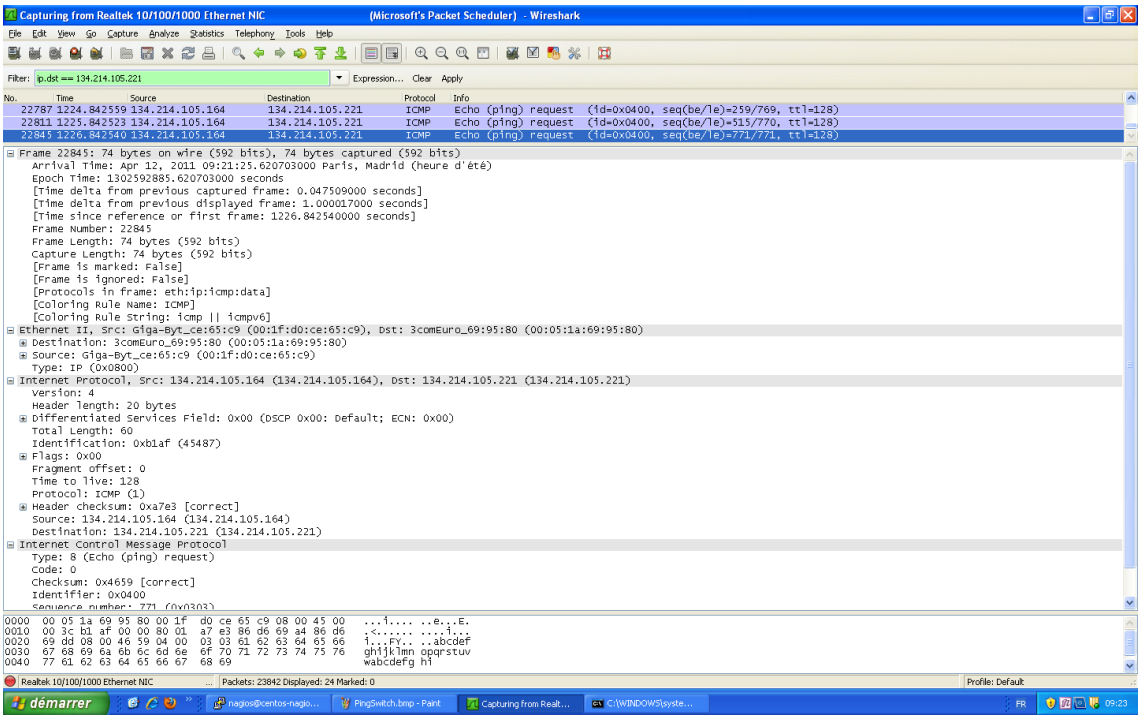


FIGURE 28 – Analyse de trame

FIGURE 29 – MIB

On constate que le serveur snmp est lancé depuis un peu plus de 7 jours. L'upTime permet de savoir s'il y a des coupures de courant.

6.4.8 Traceroute

Principe de fonctionnement : Traceroute envoie des paquets UDP avec un paramètre TTL (Time To Live) de plus en plus grand. Lorsqu'un routeur récupère un paquet ip, il décrémente TTL et retransmet le paquet. Lorque le TTL atteint 0, un paquet ICMP d'erreur Time To Live exceeded et envoyé par le

FIGURE 30 – snmpget

routeur à la source. Les routeurs sont ainsi découvert de proche en proche. Lorsque la destination est atteinte par un paquet, la source ne recoit plus de paquet ICMP d'erreur mais un paquet ICMP Port Unreachable ayant pour adresse source, l'adresse de l'élément sondé.

La commande Traceroute permet ainsi de savoir par où passe les paquets à destination d'une machine distante. Cependant, traceroute ne montre que le chemin allant de la source à la destination. En effet, le chemin de retour peut être différent. Afin de déterminer la topologie du réseau, il convient donc de faire un traceroute depuis la machine Windows vers Iftpserv2 puis depuis Iftpserv2 vers la machine Windows.

A noter que l'identification de la topologie peut ne pas être complète puisque certains routeurs peuvent ne pas répondre aux requête ICMP.

Windows – Iftpserv2

```
C:\>tracert iftpserv2

Détermination de l'itinéraire vers iftpserv2.insa-lyon.fr [134.214.104.18]
avec un maximum de 30 sauts :

  1     9 ms     2 ms     9 ms  134.214.105.1
  2    <1 ms    <1 ms    <1 ms  iftpserv2.insa-lyon.fr [134.214.104.18]

Itinéraire déterminé.
```

FIGURE 31 – tracert Iftpserv2

Iftpserv2 – Windows

```
[jdepotter1@iftpserv2 ~]$ traceroute 134.214.105.165
traceroute to 134.214.105.165 (134.214.105.165), 30 hops max, 60 byte packets
 1  ifswitch01.insa-lyon.fr (134.214.104.1)  2.922 ms  2.915 ms  2.884 ms
 2  if213-06.insa-lyon.fr (134.214.105.165)  0.337 ms  0.268 ms  0.197 ms
```

FIGURE 32 – traceroute 134.214.105.165

Topologie du réseau

Les deux traceroutes nous permettent de déduire la topologie du réseau. La machine Windows et iftpserv2 sont interconnecté par l'intermédiaire du routeur ifswitch01.insa-lyon.fr. Celui ci à au moins deux interfaces : 134.214.104.1 et 134.214.105.1.

FIGURE 33 – Topologie du reseau

6.5 Annexe 4 - Organisation de NAGIOS

Parmi les logiciels libre de supervision, Nagios est le plus répandu et également le plus suivi par la communauté de développeur. Nagios (anciennement appelé Netsaint) est une application permettant la surveillance système et réseau. Elle surveille les hôtes et services spécifiés, alertant lorsque les systèmes vont mal et quand ils vont mieux.

6.5.1 NAGIOS

Nous allons ajouter la machine windows if213-06 , IP 134.214.105.165. On va modifier les fichies de configuration : en premier le fichier NSC.ini de NSClient++ et après Nagios.cfg et Windows.cfg sur le serveur Nagios (à travers une connexion SSH). En fait on va créer un fichier de configuration (Exemple.cfg) pour chaque host qu'on ajoute et va ajouter une ligne donnant le nom de ce fichier dans Nagios.cfg : cfg_file=/usr/local/nagios/etc/objects/Exemple.cfg

NSC.INI

```
[modules]
FileLogger.dll
CheckSystem.dll
CheckDisk.dll
NSClientListener.dll
NRPEListener.dll
SysTray.dll
CheckEventLog.dll
CheckHelpers.dll

;CheckWMI.dll
[Settings]
;# OBFUSCATED PASSWORD
; This is the same as the password option but here you can store the
   password in an obfuscated manner.
; *NOTICE* obfuscation is *NOT* the same as encryption, someone with
   access to this file can still figure out the
; password. Its just a bit harder to do it at first glance.
;obfuscated_password=Jw0KAUUdXlAAUwASDAAB
;
;# PASSWORD
; This is the password (-s) that is required to access NSClient
   remotely. If you leave this blank everyone will be able to access
   the daemon remotly.
password=passe
;
;# ALLOWED HOST ADDRESSES
; This is a comma-delimited list of IP address of hosts that are
   allowed to talk to the all daemons.
```

```
; If leave this blank anyone can access the deamon remotly (NSClient
   still requires a valid password).
allowed_hosts=134.214.105.165,134.214.105.156
[log]
;# LOG DEBUG
; Set to 1 if you want debug message printed in the log file (debug
   messages are always printed to stdout when run with -test)
;debug=1
;
;# LOG FILE
; The file to print log statements to
file=NSC_H4312.log
;
;# LOG DATE MASK
; The format to for the date/time part of the log entry written to file
   .
;date_mask=%Y-%m-%d %H:%M:%S
[NSClient]
;# NSCLIENT PORT NUMBER
; This is the port the NSClientListener.dll will listen to.
port=12489
```

NAGIOS.CFG

```
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/localhost.cfg
# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

WINDOWS.CFG

```
##############################################################################
##############################################################################
#
# HOST DEFINITIONS
#
##############################################################################
##############################################################################
# Define a host for the Windows machine we'll be monitoring
# Change the host_name, alias, and address to fit your situation
define host{
use windows-server ; Inherit default values from a template
host_name if213-06 ; The name we're giving to this host
alias Machine-test ; longer name associated with the host
address 134.214.105.165 ; IP address of the host
}
##############################################################################
```

```
##############################################################################
#
# HOST GROUP DEFINITIONS
#
##############################################################################
##############################################################################
# Define a hostgroup for Windows machines
# All hosts that use the windows-server template will automatically be
   a member of this group
define hostgroup{
hostgroup_name windows-servers ; The name of the hostgroup
alias Windows Servers ; Long name of the group
}
##############################################################################
##############################################################################
#
# SERVICE DEFINITIONS
#
##############################################################################
##############################################################################
# Create a service for monitoring the version of NSCLient++ that is
   installed
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description NSClient++ Version
check_command check_nt!CLIENTVERSION
}
# Create a service for monitoring the uptime of the server
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description Uptime
check_command check_nt!UPTIME
}
# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description CPU Load
check_command check_nt!CPULOAD!-l 5,80,90
}
# Create a service for monitoring memory usage
```

```
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description Memory Usage
check_command check_nt!MEMUSE!-w 80 -c 90
}
# Create a service for monitoring C:\ disk usage
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description C:\ Drive Space
check_command check_nt!USEDDISKSPACE!-l c -w 80 -c 90
}
# Create a service for monitoring the W3SVC service
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description W3SVC
check_command check_nt!SERVICESTATE!-d SHOWALL -l W3SVC
}
# Create a service for monitoring the Explorer.exe process
# Change the host_name to match the name of the host you defined above
define service{
use generic-service
host_name if213-06
service_description Explorer
check_command check_nt!PROCSTATE!-d SHOWALL -l Explorer.exe
}
```

Résultats NAGIOS

On va commencer par monter la machine que nous avons ajouté : if213-06.

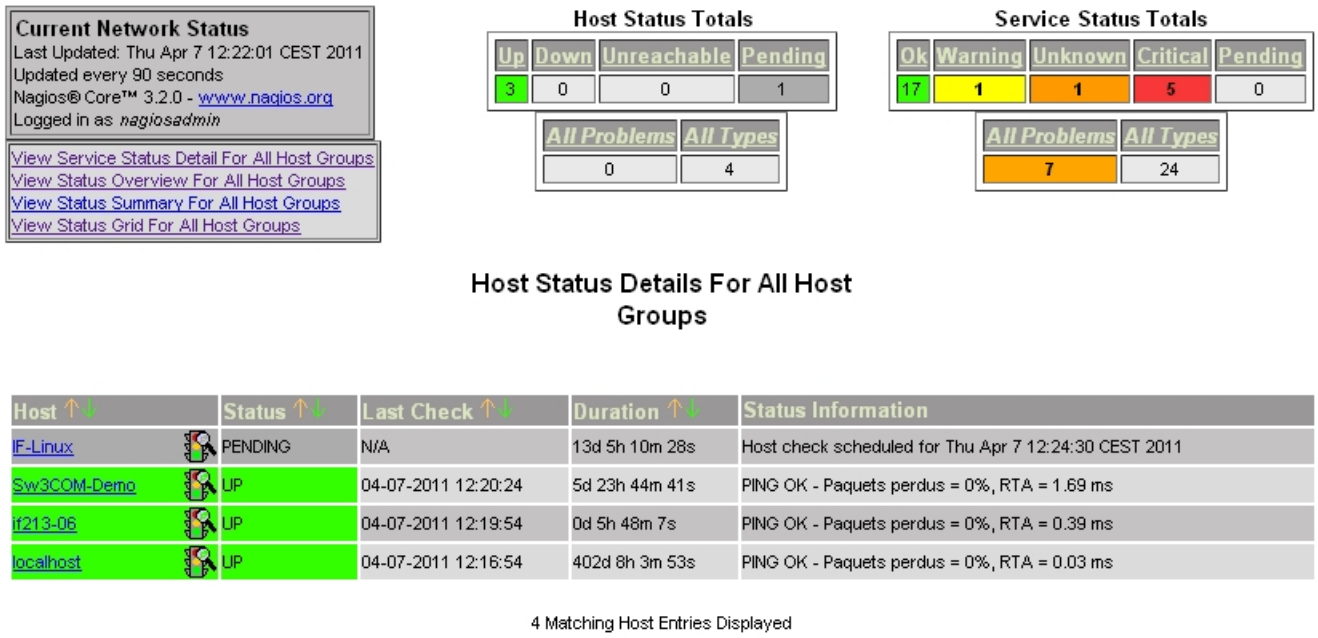


FIGURE 34 – Statut de la machine que nous avons ajouté

Localhost est la machine propre à Nagios.

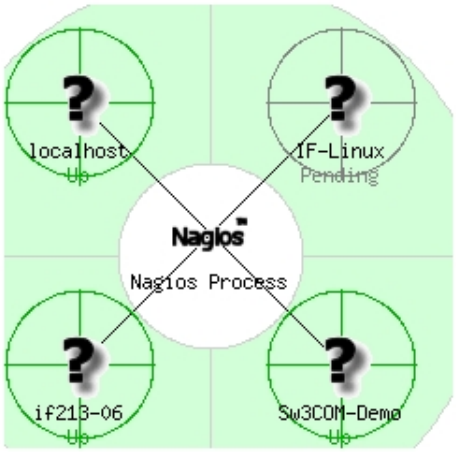


FIGURE 35 – Carte des machines surveillées dressée par Nagios

On peut aussi voir les details d'etat de chaque host et faire des check manuellement :

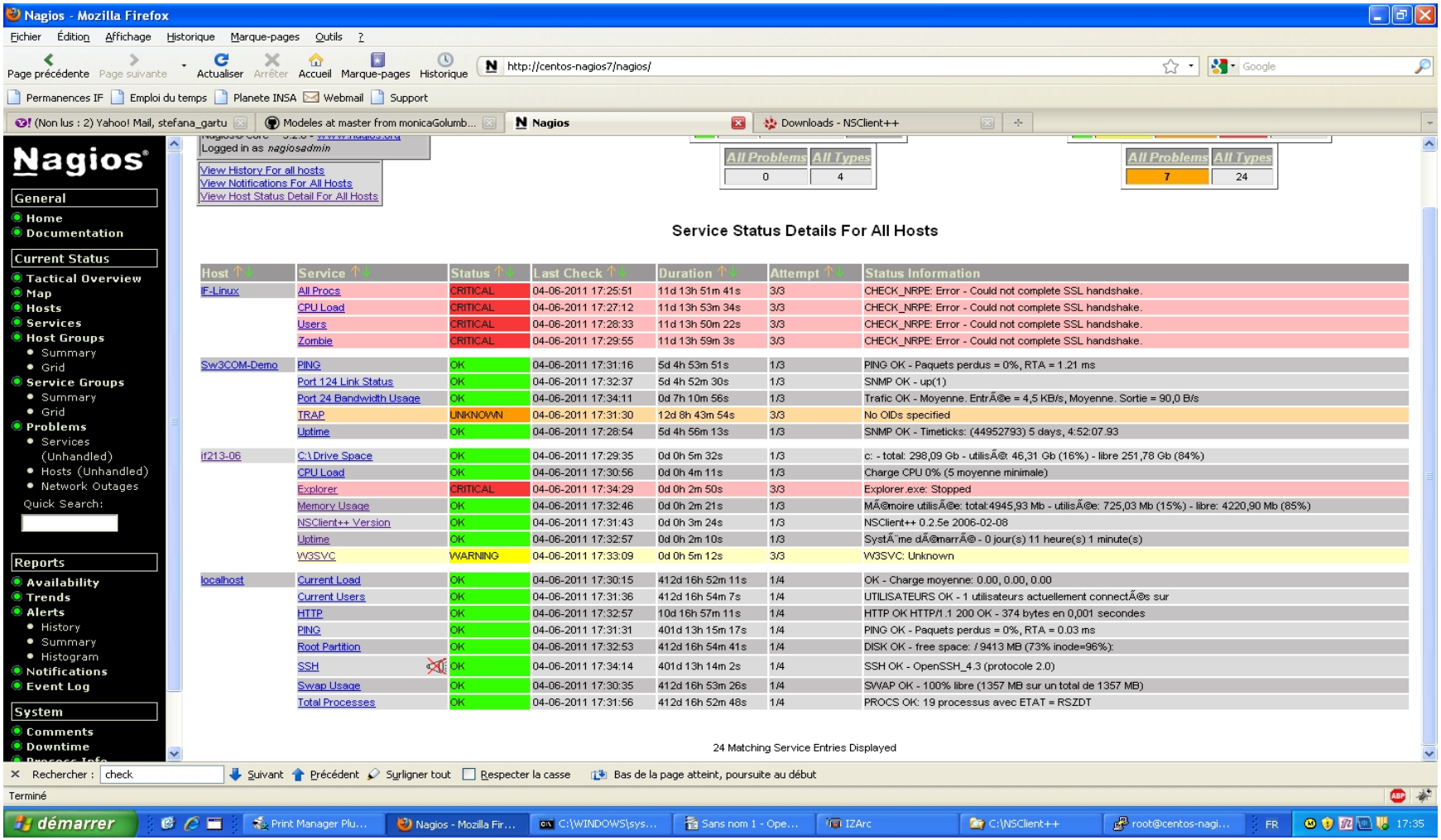


FIGURE 36 – Statuts des machines observées

On va essayer eteindre la machine if213-06, faire un check manuellement et regarder a partir d'une autre machine son status :

FIGURE 37 – Statuts après extinction de if-213-06

On peut voir que tous les services sont passé à critucal d'ou on a bien detecté que la machine à etait eteinte.

En suite on va regarder les problemes, les raports et les alerts.

Pour voir les problemes au moment actuel on peut cliquer sur Problems.

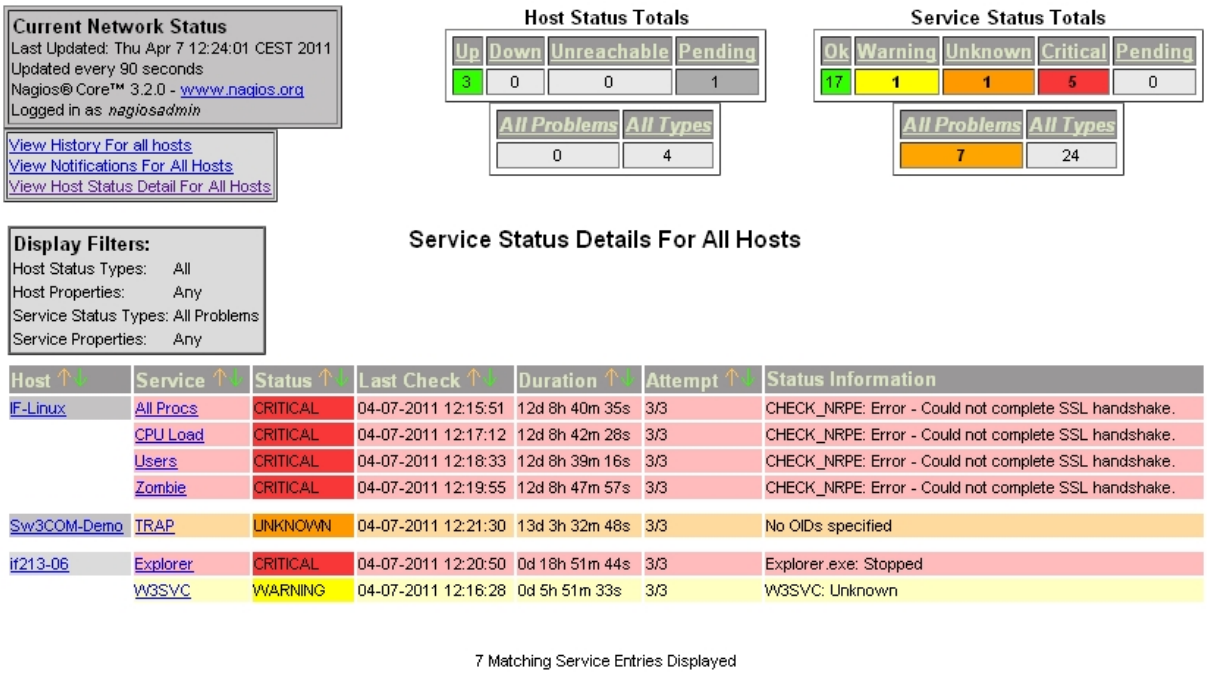


FIGURE 38 – Problèmes

Pat contre, si on veut avoir de statistiques au consulter l'historique, on peut le faire dans la partie Reports. On va créer un raport pour la derniere semaine et regarder l'activité de tous les hosts :

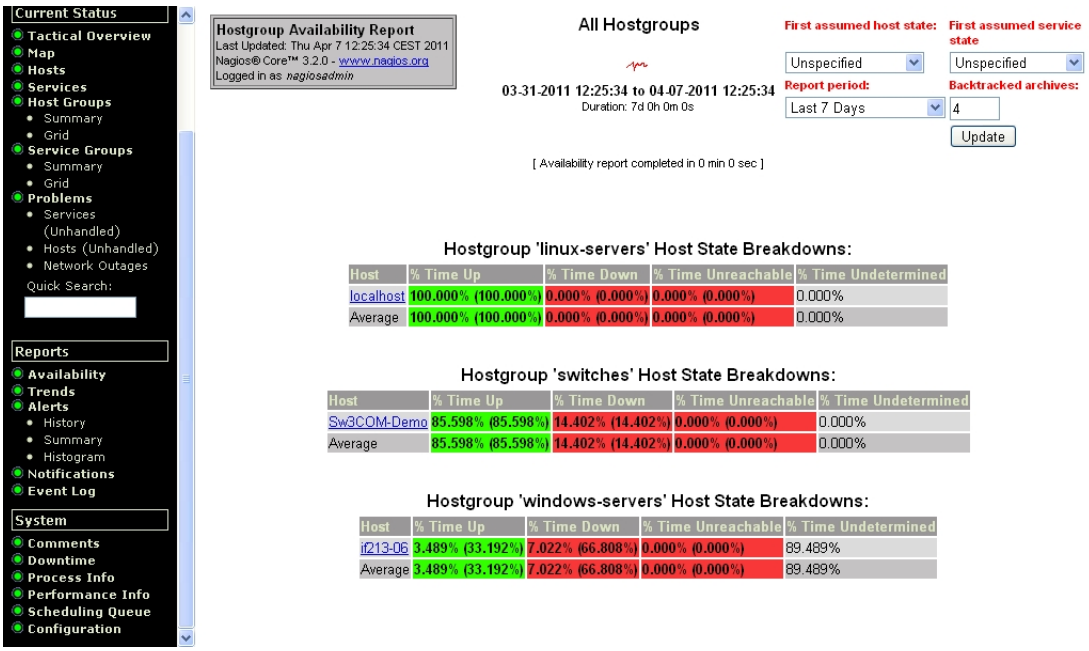


FIGURE 39 – Activité des machines par groupe

On peut aussi regrader l'activité plus en detail d'un seul host. Pour example notre host, if213-06 pendant les dernieres 24h :

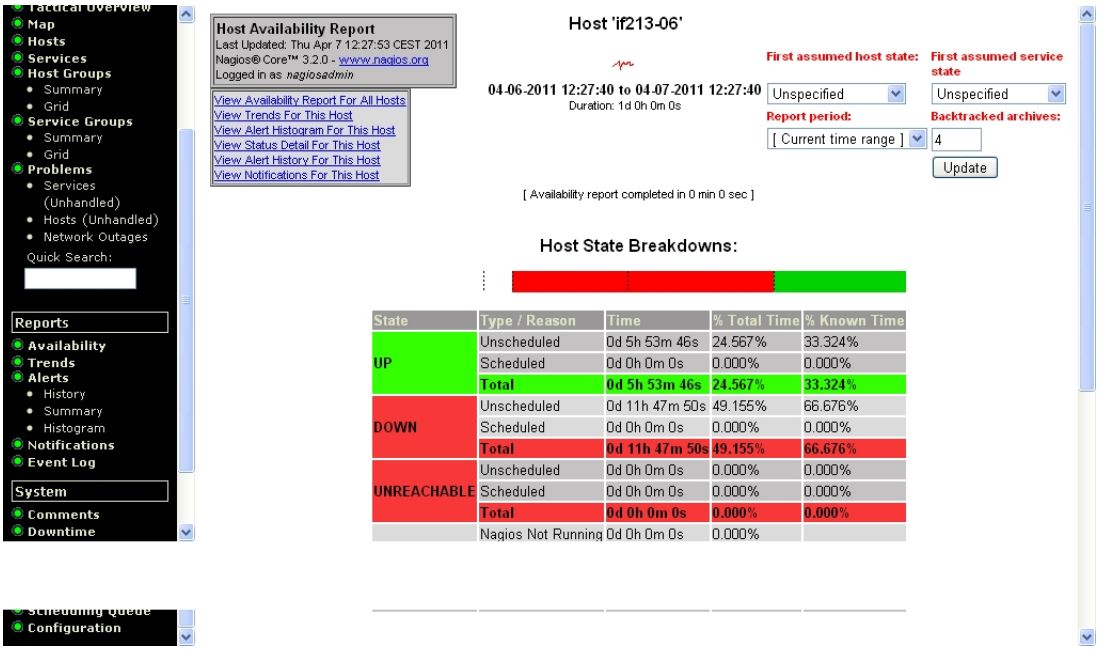


FIGURE 40 – Activité d'une machine

6.5.2 MRTG

On va monitorer le switch 134.214.105.222.

Dans un premier temps on va créer le dossier /var/www/mrtg/134.214.105.222

```
sudo mkdir /var/www/mrtg/A.B.C.D
```

On ajoute la machine a superviser et donc on crée son fichier de configuration (un fichier de configuration/machine a superviser)

```
cfgmaker --global 'WorkDir: /var/www/mrtg/134.214.105.222'
                --ifdesc=descr --global 'Language: french'
                --global 'Options[_]: bits,growright'
                public@134.214.105.222 >
                /var/www/mrtg/134.214.105.222/134.214.105.222.cfg
```

Générer automatiquement les graphes : nous allons donc créer un script SHELL que nous lancerons en utilisant crontab. Ce script SHELL va lancer MRTG avec le fichier de configuration créé auparavant.

```
#!/bin/sh
/var/www/mrtg/134.214.105.222 env LANG=C
/usr/bin/mrtg/var/www/mrtg/134.214.105.222/134.214.105.222.cfg
```

Puis on lui donne les droits en exécution :

```
chmod a+x /usr/local/bin/mrtgcron.sh
```

Et enfin, on programme la crontab (crontab -l) en ajoutant la ligne suivante :

```
*/5 * * * * /usr/local/bin/mrtgcron.sh
```

134.214.105.222.cfg

Pour pouvoir voir et afficher les graphes il faut ajouter une entrée pour chaque graphe dans le fichier de configuration 134.214.105.222.cfg (on ne mettra ici que le code pour le premier graphe, les 3 autres étant similaires).

```
# /usr/bin/cfgmaker --global 'WorkDir: /var/www/mrtg/134.214.105.222'
#                   --ifdesc=descr --global 'Language: french'
#                   --global 'Options[_]: bits,growright'
#                    public@134.214.105.222
### Global Defaults
EnableIPv6: no
WorkDir: /var/www/mrtg/134.214.105.222
Language: french
Options[_]: bits,growright
### Interface 101 >> Descr: 'RMON-Port-01-on-unit-1' | Name: '' | Ip:
    '' | Eth: '' ###
Target[134.214.105.222_101]: 101:public@134.214.105.222:
SetEnv[134.214.105.222_101]: MRTG_INT_IP="" MRTG_INT_DESCR="RMON-Port
    -01-on-unit-1"
MaxBytes[134.214.105.222_101]: 12500000
Title[134.214.105.222_101]: RMON Port 01 on unit 1 -- switchTP222
PageTop[134.214.105.222_101]: <h1>RMON Port 01 on unit 1 -- switchTP222
    </h1>
<div id="sysdetails">
<table>
<tr>
<td>System:</td>
<td>switchTP222 in chez nous</td>
</tr>
<tr>
<td>Maintainer:</td>
<td>admin</td>
</tr>
<tr>
<td>Description:</td>
<td>RMON-Port-01-on-unit-1 RMON Port 01 on unit 1 </td>
</tr>
```

```
<tr>
<td>ifType:</td>
<td>ethernetCsmacd (6)</td>
</tr>
<tr>
<td>ifName:</td>
<td></td>
</tr>
<tr>
<td>Max Speed:</td>
<td>100.0 Mbits/s</td>
</tr>
</table>
</div>
```

Résultats MRTG

On a monitoré pendant 3 jours. Dans le repertoire www/mrtg/134.214.105.222 on a 16 images generées automatiquement. On a des information par jour, semaine, mois, an pour les 4 ports choisis.

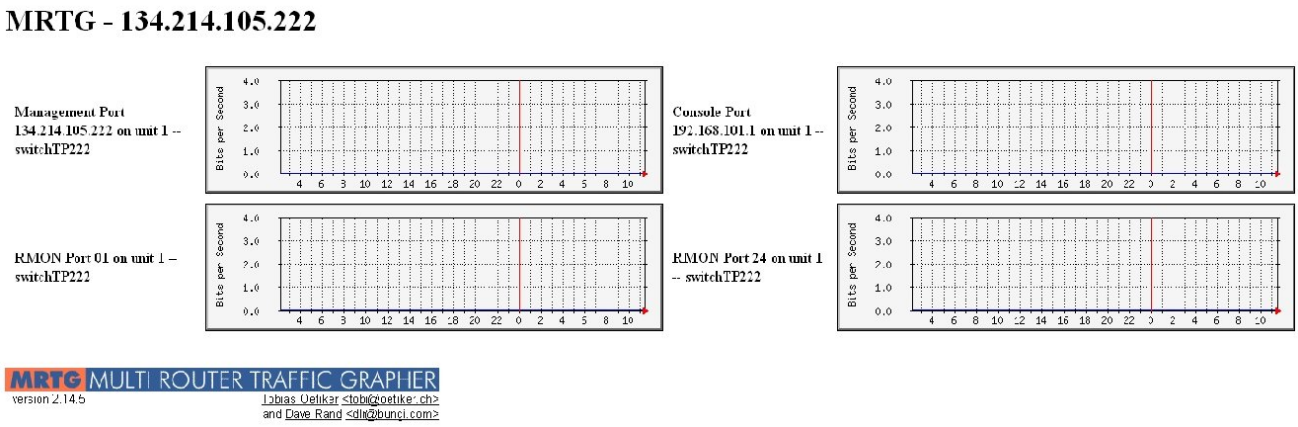


FIGURE 41 – Graphiques une heure après démarrage

Après 3 jours pour le Managemet Port, 134.214.105.222 on unit 1, swtchTP222 :

Image par jour :

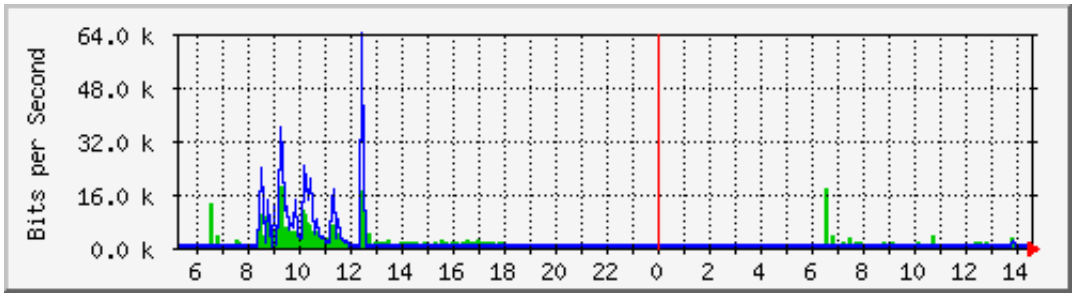


FIGURE 42 – Image par jour

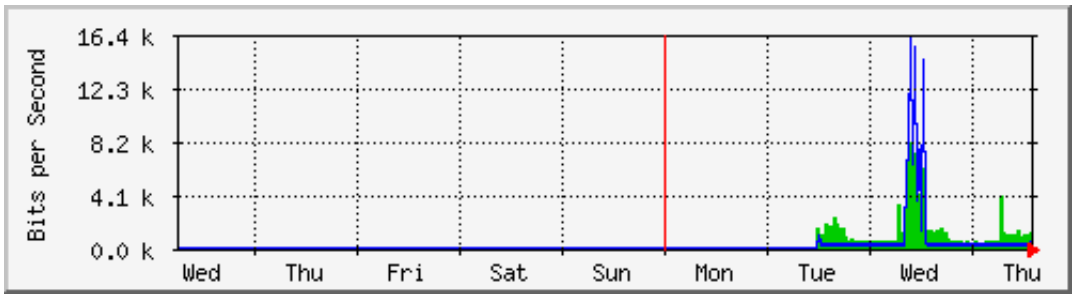


FIGURE 43 – Image par semaine

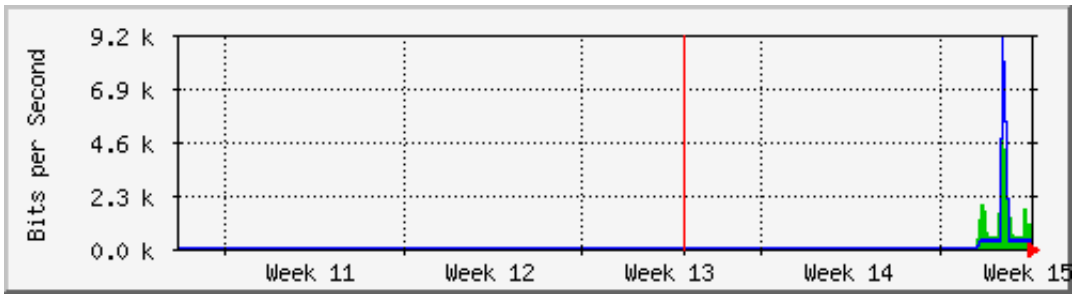


FIGURE 44 – Image par mois

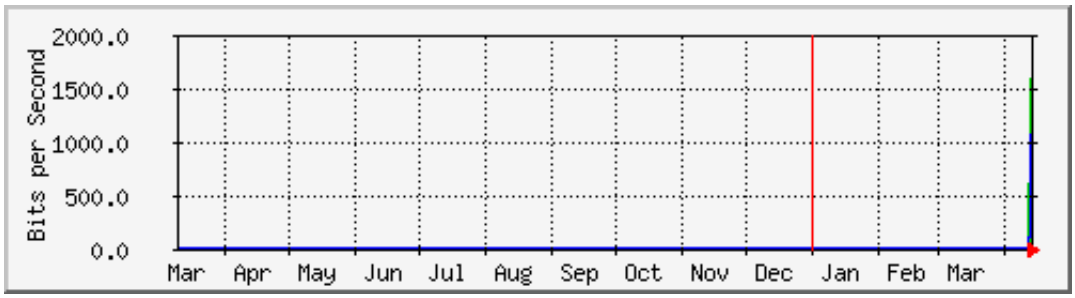


FIGURE 45 – Image par an

6.5.3 NRPE

A l'aide de NRPE nous pouvons superviser des machines Linux. On a par exemple ajouté la machine 134.214.105.189 (nom : h4312_linux) et on a changé la configuration du serveur Nagios pour qu'il utilise NRPE (nouveau nom : h4312_localhost).

On a repris la base NRPE qui était fournie et on l'a modifié pour pouvoir ajouter les deux machines.

NRPE.CFG

```
# LOG FACILITY
# The syslog facility that should be used for logging purposes.
log_facility=daemon
# PID FILE
```

```
# The name of the file in which the NRPE daemon should write it's
    process ID
# number. The file is only written if the NRPE daemon is started by the
     root
# user and is running in standalone mode.
pid_file=/var/run/nrpe.pid
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow
    clients
# to specify arguments to commands that are executed. This option only
    works
# if the daemon was configured with the --enable-command-args configure
     script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security
    implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments
dont_blame_nrpe=0
# DEBUGGING OPTION
# This option determines whether or not debugging messages are logged
    to the
# syslog facility.
# Values: 0=debugging off, 1=debugging on
debug=0
# COMMAND TIMEOUT
# This specifies the maximum number of seconds that the NRPE daemon
    will
# allow plugins to finish executing before killing them off.
command_timeout=60
# CONNECTION TIMEOUT
# This specifies the maximum number of seconds that the NRPE daemon
    will
# wait for a connection to be established before exiting. This is
    sometimes
# seen where a network problem stops the SSL being established even
    though
# all network sessions are connected. This causes the nrpe daemons to
# accumulate, eating system resources. Do not set this too low.
connection_timeout=300
# COMMAND DEFINITIONS
# Command definitions that this daemon will run. Definitions
# are in the following format:
```

```
#
# command[<command_name>]=<command_line>
#
# When the daemon receives a request to return the results of <
    command_name>
# it will execute the command specified by the <command_line> argument.
#
# Unlike Nagios, the command line cannot contain macros - it must be
# typed exactly as it should be executed.
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c
    30,25,20
command[check_hda1]=/usr/local/nagios/libexec/check_disk -w 20% -c 10%
    -p /dev/sda1
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5
    -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150
     -c 200
command[check_hdd]=/usr/local/nagios/libexec/check_disk -w 20 -c 10 -p
    /
command[check_swaphdd]=/usr/local/nagios/libexec/check_swap -w 20 -c 10
```

h4312_linux.cfg

On a ajouter les services demandés (gestion des users, zombies, processus et charge CPU) pour monitorer la machine. Le code pour le serveur Nagios est similaire a celui-ci et donc on va pas le mettre dans le compte-rendu.

```
###############################################
# Monitoring teacher's linux avec NRPE
###############################################
############################
## HOST
############################
define host {
use linux-server
host_name h4312_linux
alias remote Linux
address 134.214.105.189
}
############################
## SERVICES
############################
# Charge CPU
define service{
use generic-service
host_name h4312_linux
```

```
service_description CPU Load
check_command check_nrpe!check_load
}
# Users
define service{
use generic-service
host_name h4312_linux
service_description Users
check_command check_nrpe!check_users
}
# Zombies
define service{
use generic-service
host_name h4312_linux
service_description Zombies
check_command check_nrpe!check_zombie_procs
}
# Processus
define service{
use generic-service
host_name h4312_linux
service_description Processus
check_command check_nrpe!check_total_procs
}
```

NAGIOS.CFG

```
# Definitions for monitoring the local (Linux) host
cfg_file=/usr/local/nagios/etc/objects/h4312_localhost.cfg
# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

6.5.4 Résultats NRPE

On a ajouté la machine linux monitorée a l'aide de NRPE :

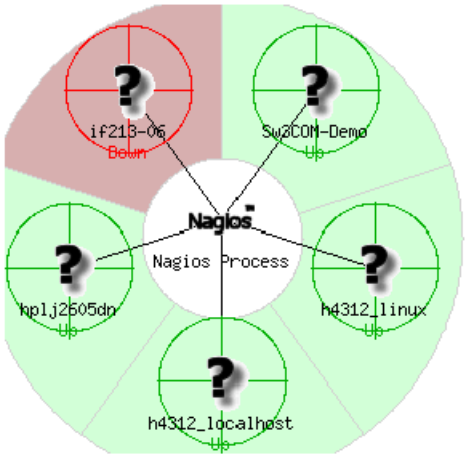


FIGURE 46 – Machines monitorées par noagios

Le client sur la machine if213-06 est éteint. On peut visualiser pour la machine Linux le status des parametres qu'on a ajouté :

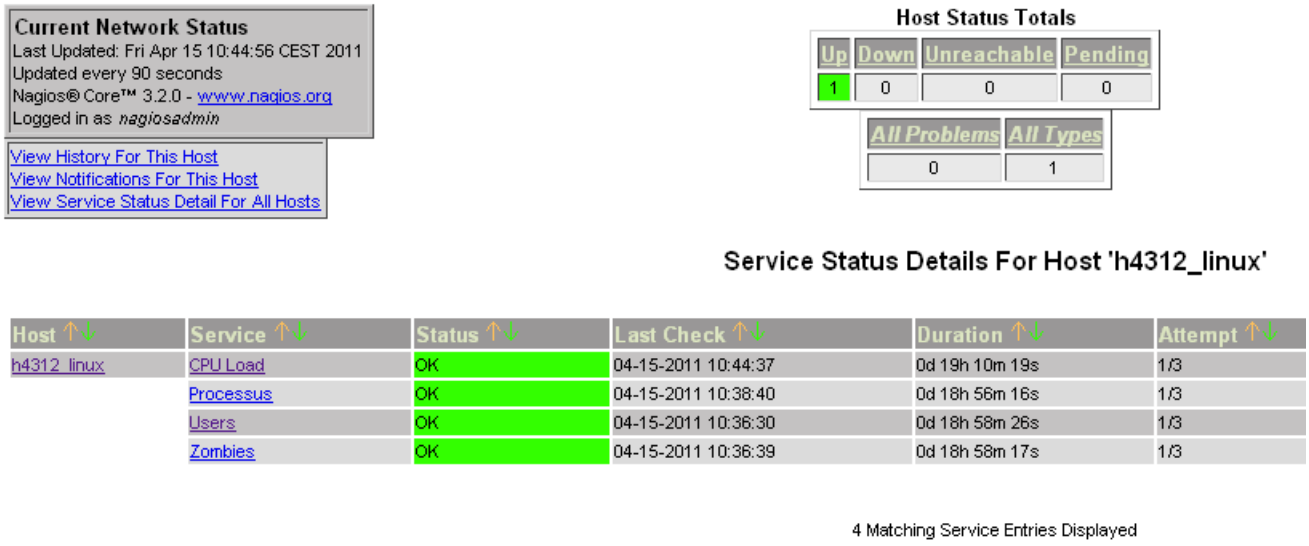


FIGURE 47 – Statuts des paramètres observés

6.5.5 Analyse critique de l'Installation/Utilisation de Nagios

Le gros avantage de Nagios est le fait que c'est un outil qui dispose d'une documentation très fournie. Il est très modulable et flexible grâce à son fonctionnement par fichier de configuration. Mais ce qui fait sa force fait aussi sa faiblesse car ces fichiers de configurations multiples peuvent poser des problème s'il y en a beaucoup, particulièrement lorsqu'on n'est pas habitué à cette méthode de fonctionnement. Cela nécessite une bonne organisation pour s'y retrouver mais lorsqu'on est bien organisé, on peut facilement automatiser le traitement des informations grâce à des scripts.

Son installation n'est pas très compliquée. C'est un simple système client-serveur. On installe le serveur puis les clients puis il faut configurer. Il faut configurer les clients pour qu'ils acceptent les connexions et

les échanges d'information. Il faut configurer le serveur pour chaque client que l'on ajoute. Il faut savoir que l'on peut utiliser un seul fichier de configuration pour tout le programme mais ce n'est pas conseillé pour une raison de lisibilité. Il faut donc bien réfléchir à l'arborescence des fichiers dès le départ, ce qui peut être difficile lorsque l'on n'a pas l'habitude.

Son utilisation par l'interface web est assez intuitive et permet une navigation rapide entre les différents services associés aux machines monitorées. La maintenance (ajout de nouvelles machines, nouveau services, ...) dépend beaucoup de la façon dont ont été créé les fichiers de configuration au départ. Cela peut donc varier entre très simple, voire automatisé et très compliqué. Pour nous, une fois que nous avons compris comment tout s'agençait, cela a été plutôt facile.

C'est un outil qui permet de regrouper les informations sur toutes les machines surveillées, ce qui est très pratique mais il ne fait que du monitoring système et ne permet pas le monitoring réseau. Il ne reconnaît pas non plus la topologie réseau, ce qui peut poser quelques problèmes, notamment lorsque les machines sont équipées d'adresses IP dynamiques (les fichiers de configurations ne peuvent pas prendre en compte ce type de cas). Ce pourrait être un outil très puissant couplé avec une application de monitoring réseau.