REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE.

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE
D'ALGER
DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention du
Diplôme d'Ingénieur d'Etat en Electronique

Thème :

Mise en place d'une plate-forme de Métrologie passive pour le réseau d'un campus universitaire

Réalisé par :

OUDJEHANE Said
SAHRAOUI Kamel Lamine

proposé par : **Mr. SADOUN Rabah**

Membres du jury :

Pr BELOUCHRANI	**Président**
Mr SADOUN	**Rapporteur**
Pr TRABELSI	**Examinateur**

-Promotion Juin 2009-

Remerciements

On tient à exprimer nos sincères remerciements à notre promoteur Mr. Rabah Sadoun pour nous avoir proposé ce sujet, guidé tout au long de notre parcourt avec ses précieux conseils et sa confiance ;

On remercie notamment, les membres du jury à savoir : Mr Mohamed Trabelsi et Mr Adel Belouchrani pour l'honneur qu'ils nous font en acceptant d'être les évaluateurs et les rapporteurs de ce sujet ;

On remercie aussi tous les enseignants de l'Ecole Nationale Supérieure Polytechniques pour leur gentillesse et le savoir faire qu'ils nous ont transmis durant nos cinq années universitaires ;

On oublie pas de remercier nos parents qui ont veillé sur nous jusqu'à aujourd'hui, car sans eux nous n'aurions pas atteint ce stade ;

Enfin, on remercie tout les gens qui ont contribué de prés ou de loin à l'achèvement de ce projet, spécialement l'équipe du centre de calcul.

Par Oudjehane Said

Sahraoui Kamel Lamine

Dédicaces

Je dédie cet humble travail à tous ceux qui sont chers à mon cœur :

- *D'abord à mes très chers parents qui ont tout prodigué pour que j'arrive là où j'en suis ;*
- *A mes frères adorés : Walid, Mounir, Amine ;*
- *A Amou Rachid qui a beaucoup veillé sur ma carrière d'étudiant avec ses précieux conseils ;*
- *Sans oublier Khouya Rafik qui m'a beaucoup soutenu et à sa femme Soumeya, son frère Reda, ses sœurs : Hayet et Wided et à leurs familles ;*
- *A ma très chère grand'mère Mamia que j'adore ;*
- *A mes tentes Zoubida et Akila et mon oncle Boualem sans oublier mes cousins : Bilel, Mounia, Redouane, Sarah, Lilia, Samir, Soufiane ;*
- *A Amou Mouhoub et à tous les membres de la famille Hammeniche ;*
- *A tous mes amis intimes : Abdelhak (Alukard), Oussama (Dracula), Nabil, Redouane, Hakim, Madjid, baya et spécialement Feriel.*
- *A mon binôme KamelAmine ;*

A toute la promotion 2009.

Par Oudjehane Said

Je dédie ce modeste travail :

- *À Mes parents ;*
- *À Mon frère et ma sœur ;*
- *À Toute la famille Sahraoui et Bacha ;*
- *À Mon binôme Said, tous mes amis de la promotion 2009 et particulièrement mes amis d'électronique ;*

Par Sahraoui Kamel Lamine

Table des matières

Chapitre II: Mesures Passives via SNMP

Chapitre III: Monitoring des réseaux IP

Chapitre IV: Mise en place de la plate-forme de Métrologie passive

Liste des figures

Liste des tableaux

Table des abréviations

ACK	Acknowledge
ASN.1	Abstract Syntax Notation
CMIP	Common Management Information Protocol
CPU	Central Process Unit
EGP	Exterior Gateway Protocol
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
IPPM	IP performance Metrics
HTTP	Hypertext Transfer Protocol
LAN	Local Area Network
MIB	Management Information Base
MRTG	Multi Router Traffic Grapher
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
OID	Object Identifier
RFC	Request For Comment
RRDTool	Round Robin Database Tools
RTT	Round to Trip
SEC	Seuil d'évitement de congestion
SMI	Structure Of Management Information
SNMP	Simple Network Management Protocol
TCP	Transport Control Protocol
TNM	Telecommunications Network Management
UDP	User Datagram Protocol
WAN	Wide Area Network

Introduction générale

Jusqu'à 1994, la tendance courante dans les réseaux, comme dans Internet, était d'agrandir l'infrastructure au fur et à mesure que le trafic de données augmentait. Dans cette perspective, les architectes et administrateurs réseaux étaient préoccupés plus par des problèmes d'interconnexions et d'interopérabilité que de calibrage et d'analyse de tendances des réseaux.

Mais depuis 1995, l'état d'Internet a considérablement évolué (décentralisation de l'administration, communautés d'utilisateurs avec des exigences différentes, introduction de nouveaux services) rendant nécessaire l'obtention d'informations sur son comportement et ses performances.

Les différentes communautés (utilisateurs, communauté scientifique, fournisseurs de service, commerciaux et vendeurs) ont alors pris conscience de l'importance de la métrologie des réseaux, en tant qu'élément clé dans le développement d'une infrastructure robuste, fiable et performante.

A une moindre échelle, l'objectif est le même pour tous les administrateurs de réseaux : connaitre et comprendre son réseau afin de pouvoir, non seulement intervenir dans l'urgence en cas de problème, mais aussi anticiper sur son évolution, planifier l'introduction de nouvelles applications, et améliorer ses performances. Dans cette optique, les mesures constituent également un élément clé de la gestion du réseau.

Dans le contexte de notre projet, on a réalisé une plate-forme de métrologie passive, proposant des services de gestion et de suivie des performances, pour un réseau d'un campus universitaire en exploitant des outils appropriés à cette tache.

Ce rapport est structuré comme suit : dans le chapitre I, on a présenté de manière générale le domaine de la Métrologie des réseaux. On s'est intéressé au cas particulier des mesures passives. On a abordé en détail dans le chapitre II, le protocole SNMP (Simple Network Management Protocol), utilisé pour la récolte d'informations sur le réseau. On a montré dans le chapitre III, comment ces informations récupérées peuvent être présentées et mises en valeur par le Monitoring via l'outil MRTG. Ceci nous a permis, dans le chapitre IV, de mettre en œuvre notre application en utilisant l'ensemble de ces outils.

CHAPITRE I

Introduction à la Métrologie des Réseaux

Introduction

La métrologie est une activité essentielle pour le bon fonctionnement de tout type de réseau. Elle met en valeur les contraintes dues à la latence et aux pertes de données. Elle recouvre des domaines d'étude comme :

- La classification du trafic pour pouvoir trier les flux en fonction de la qualité de service (QoS) qu'ils requièrent.
- Le dimensionnement des réseaux, qui permet de mettre en place des dispositions suffisantes, pour assurer en permanence un service adéquat à tous les utilisateurs.
- L'analyse des mécanismes de contrôle du réseau tels que les algorithmes de routages et de transport qui assurent le traitement du flux, ainsi que des erreurs et de la congestion, etc.

Cette analyse donne l'accès à la compréhension de tous ces mécanismes qui interagissent et permet ainsi de régler d'une façon fine les différents paramètres à prendre de compte.

Nous aborderons dans ce qui suit les notions importantes dont il faut tenir compte lors de l'administration et la supervision d'un réseau.

1. Classification du trafic réseau

On trouve deux grandes classes de trafic :

- Le trafic « streaming » est typiquement produits par les services téléphoniques et vidéo. Le délai de transfert des données et la gigue doivent être contrôlables dans ce type de trafic, tandis qu'un certain degré peut être tolérable pour la perte de paquets.
- Le trafic dit « élastique », ainsi nommé car son débit peut s'adapter à des contraintes extérieures (bande passante insuffisante par exemple) sans pour autant remettre en cause la viabilité du service. Cette classe de trafic est essentiellement engendrée par le transfert d'objets numériques tels que des pages Web (application HTTP), des messages électroniques (E-mail, application SMTP) ou des fichiers de données (application FTP). Le respect de leur intégrité est indispensable mais les contraintes de délai de transfert sont moins fortes.

Le trafic « élastique » utilise le protocole TCP alors que le trafic streaming utilise UDP. Actuellement le trafic de type streaming est en forte augmentation par rapport à TCP surtout avec l'apparition des sites internet dédiés aux partagent des vidéos.

2. Les métriques réseaux

2.1 Définition

Une métrique nous permet d'évaluer une mesure dans le but de caractériser le comportement d'un réseau.

2.2 Classification des métriques réseaux

On distingue 3 types de métriques :

2.2.1 Métrique de perte

Les medias continus (audio et vidéo) ont comme caractéristiques d'être plus ou moins redondants. Ainsi deux images successives d'une transmission vidéo comportent généralement peu de différences.

De cette redondance résulte la possibilité que des pertes d'information soient acceptables du point de vue de l'utilisateur final. Il apparait pour les medias, les applications multimédia (audio et vidéo), une contrainte de fiabilité du transfert des données non plus totale mais partielle, la perte de certaines informations pouvant être acceptable.

2.2.2 Métrique de débit

Les besoins des applications en terme de débit sont très variables. Certaines, comme les applications Web ou mail ne requièrent que quelques Kilo-octets de bande passante pour les flux qu'elles échangent. D'autres, au contraire, sont beaucoup plus exigeantes. Bien sûr, c'est cette dernière catégorie qui tend à se généraliser car de plus en plus d'applications récentes nécessitent une bande passante importante.

On peut citer les applications de diffusion en temps réel comme par exemple les chaînes de télévision sur Internet. Dans ce dernier cas, il est d'ailleurs primordial de pouvoir fournir le service le plus stable et le plus régulier possible, de façon à ce que l'utilisateur à l'extrémité du réseau reçoive son flux multimédia avec un bon niveau de qualité.

2.2.3 Métrique temporelle

Les applications multimédias présentent des contraintes temporelles, ces derniers s'expriment généralement par le biais de deux paramètres : le délai de transfert des données et la gigue.

- Délai :

  Pour les applications interactives (telle que la visioconférence), et à un degré moindre, pour les applications de diffusion en temps réel (telles que le streaming audio ou vidéo), afin que la communication se déroule comme si elle avait lieu localement, il faut que les données soient transmises en un temps inférieur au seuil de perception humain lié au media considéré. Il apparaît ainsi une contrainte sur le délai du transfert de bout en bout des données.

- Gigue :

Les medias continus (tels que l'audio et la vidéo) présentent des contraintes temporelles. Ces dernières s'expriment en termes de régularité dans l'arrivée des données. Les variations de délai (aussi appelées gigue) dans les réseaux locaux ou étendus perturbent le flux de paquets audio lors de la traversée, car elles transforment un flux périodique en un flux non périodique. Il est donc nécessaire de restaurer la périodicité originale à la destination. **[I.1]**

Ceci va nous amener à parler de la latence pour en savoir plus sur les causes de ces délais.

3. La latence

3.1 Définition

La latence ou temps de réponse est une contrainte importante dans un réseau. Elle est due au support de transmission utilisé et au temps d'attente.

La latence est la somme de :

- Durée de transmission : taille du message / débit

Temps nécessaire pour transmettre les données

- Temps de propagation : distance / vitesse de propagation

Temps nécessaire pour que les données aillent de l'émetteur au récepteur

- Temps d'attente : qui est le temps "perdu" par le système de communication (notamment à cause de l'occupation des ressources).

3.2 Latence et modèle OSI

Dans le cas du modèle OSI **[I.2]**, la contribution en latence de chaque couche est différente. Les couches réseau et transport génèrent le plus de latence. Cela est dû au fait que ces derniers gèrent, respectivement, le routage des paquets en fonction de la saturation des chemins, et la connexion de bout en bout à travers des algorithmes de contrôle de flux.

3.3 Cas particulier de la couche transport

La couche transport, étant celle qui consomme le plus la bande passante, il est nécessaire d'y établir des mécanismes de contrôle. Parmi eux, on cite :

3.3.1 Algorithme de contrôle du flux

Le contrôle de flux ce fait à travers le protocole TCP (voir Annexe A). Ce dernier est implémenté dans la couche transport du modèle OSI qui gère l'acheminement des paquets de bout en bout et leur restauration. Le protocole TCP permet, à travers différents algorithmes, le contrôle de flux des données du coté émetteur et récepteur et cela pour éviter la congestion du réseau et ainsi une bonne utilisation de la bande passante. L'évolution des algorithmes de contrôle a diminué de manière remarquable la latence dans les réseaux.

3.3.1.1 Principe

Le protocole TCP inclut des mécanismes de contrôle de flux qui contribue à la fiabilité des transmissions TCP en réglant le taux effectif de flux de données entre les deux services de la session. Quand la source est informée que la quantité de données spécifiée dans les segments a été reçue, elle peut continuer à envoyer plus de données pour cette session.

Ce champ Taille de fenêtre dans l'en-tête TCP précise la quantité de données pouvant être transmise. La taille de fenêtre est initialement déterminée lors du démarrage de la session.

La figure I-1 présente une représentation simplifiée de la taille de fenêtre et des reçus.

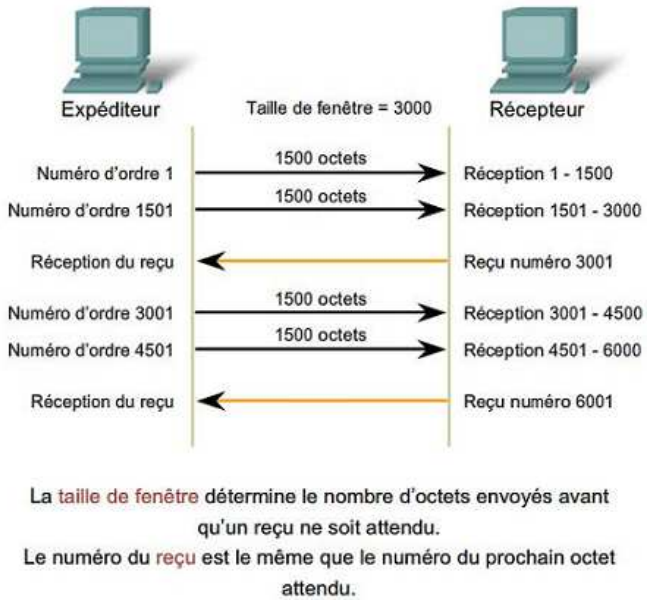


Figure I-1 : Ouverture de la fenêtre d'émission

La taille de la fenêtre d'émission est dynamique, elle est réduite si on a une perte d'un paquet dû à une saturation de la bande passante.

Quand un certain temps est écoulé sans perte de données ni contraintes excessives sur les ressources, la taille de la fenêtre continuera à augmenter jusqu'à ce que des données soient à nouveau perdues. Ces augmentations et réductions dynamiques constituent un processus qui détermine la taille de la fenêtre optimale (cf. figure I-2).

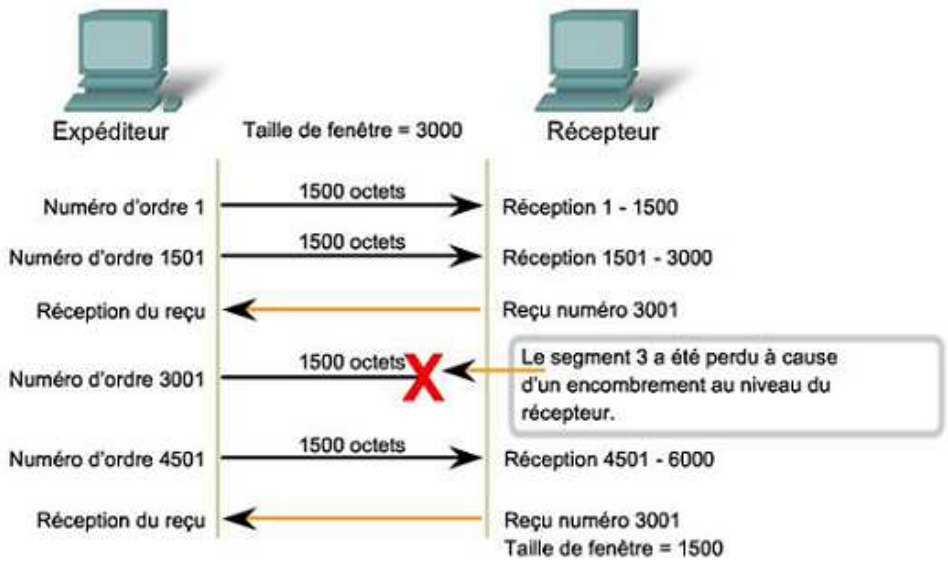


Figure I-2 : L'encombrement TCP et contrôle de flux

3.3.1.2 Structure de l'algorithme

Les algorithmes de contrôle passent généralement par 2 phases.

a) Slow Start

- l'émetteur envoie un segment et attend le ACK correspondant, quand il reçoit le ACK, l'émetteur incrémente sa Fenêtre d'Emission (FE) de 1 MSS. Il envoie donc 2 segments et attend les ACK, à chaque RTT [*].

  Si tout va bien, la FE est doublée (cf. figure I.3), donc la FE (et aussi le débit) croît exponentiellement jusqu'à un seuil prédéfini (SEC).

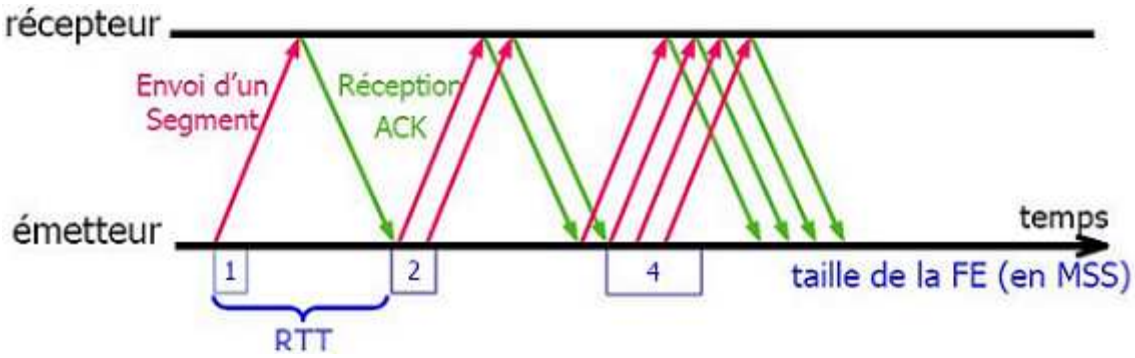


Figure I.3 : Doublement de la fenêtre d'émission

Remarque :

L'unité est donnée en MSS.

[*] RTT : le temps entre l'envoi et l'arrivé d'un accusé de réception de la donnée

b) Evitement de congestion

- Quand le seuil SEC (Seuil d'évitement de congestion) est atteint, l'émetteur passe en mode évitement de congestion.

Quand il reçoit l'ACK, l'émetteur incrémente sa FE. Pour chaque RTT, la FE est augmentée de 1MSS en respectant toujours la condition : FE < FR.

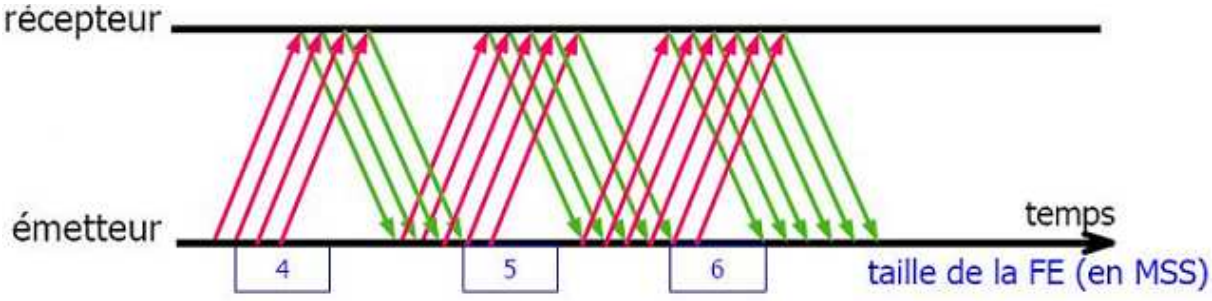


Figure I.4 : Incrémentation de la FE

En cas de congestion, l'émetteur réémet le paquet qui n'a pas été acquitté puis divise par 2 le seuil SEC et repart en mode Slow-Start (FE = 1MSS).

Exemple d'algorithme :

```
Pour chaque ACK {

Si (FE < SEC) alors {FE=2*FE}…… (Slow Start)

Sinon {FE =FE+1} ……... (Évitement de congestion)
}
Pour chaque perte de segment TCP {
SEC = max (FE/2, 2)
FE = 1 (on repart en Slow Start)
}
```

En respectant les conditions FR > bande passante*RTT et FE < FR.

La finalité de cet algorithme nous donne une FE optimale. La figure I-5 montre l'évolution de la FE dans le temps.

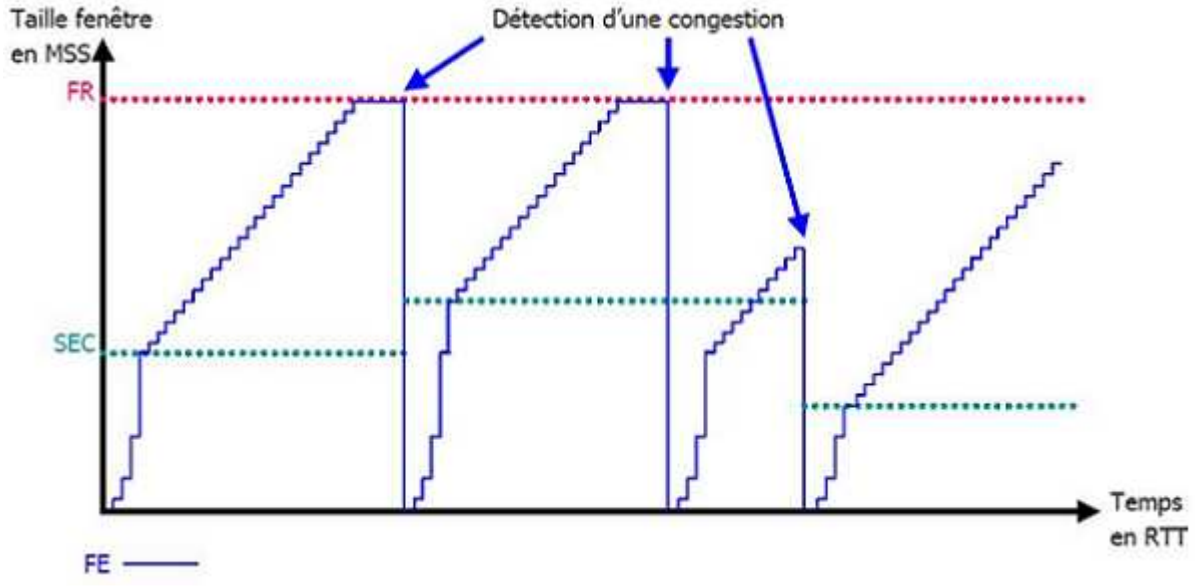


Figure I.5: Evolution de la taille de la fenêtre d'émission

Il existe différents algorithmes de contrôle, on cite :

- TCP TAHOE
- TCP RENO
- TCP NEW RENO
- TCP VEGAS

3.3.2 Débit maximum sur un lien

Le débit sur un lien n'atteint pas sa capacité maximale. Durant le passage des données sur les différentes couches du modèle OSI, elles sont encapsulées par l'ajout des en-têtes de chaque couche (en-tête TCP, en-tête IP). Ceci entrainera une occupation de la bande passante du lien par les en-têtes.

La figure I-6 montre les tailles des différents champs, relatifs aux couches du modèle OSI.

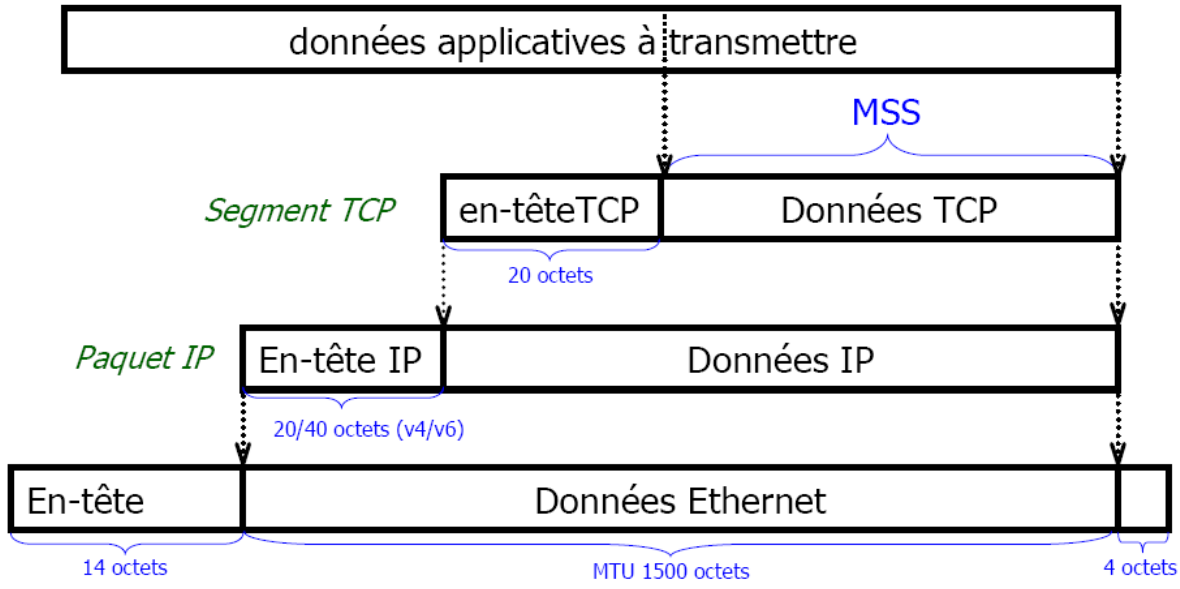


Figure I.6 Processus d'encapsulation des données

- MSS : taille maximale du segment TCP.
- Avec un MTU de 1500 octets sur Ethernet, le MSS vaut 1460 octets en IPv4 et 1440 octets en IPv6.

Sur un lien Ethernet à 100Mb/s, le débit maximal sans tenir compte des pertes éventuelles pour une trames de 1500 octets de données est de :

10**8/ (1538*8) = 8127 trames par seconde.

[1538 = 12 (inter-trame) + 8 (préambule) +14 (en-tête trame) + 1500 + 4 (CRC)]

Le débit maximal en TCP sera donc de

8127*1460 = 11,86 Mo/s (ou 94,92 Mb/s) en IPv4

8127*1440 = 11,70 Mo/s (ou 93,62 Mb/s) en IPv6

On remarque que Le protocole IPv6 consomme plus en bande passante, mais son introduction est nécessaire vu la saturation du protocole IPv4.

Remarque :

- Il est impossible d'utiliser tout la bande passante sur un lien.
- Si on a des pertes de paquets sur un réseau, la bande passante diminuera davantage.

3.3.3 Algorithme de routage

L'objectif du routage est de déterminer une route (i.e. un ensemble de liens à parcourir), en respectant certaines contraintes. Ceci dans le but d'établir une connexion entres deux nœuds : source et destinataire.

La qualité de service que l'on est en mesure d'offrir à une connexion est directement liée au choix du chemin. Le calcul de route doit prendre en compte les différentes contraintes imposées par la connexion (débit, taux de perte). Ces paramètres peuvent en effet varier en fonction des chemins empruntés.

Il faut donc utiliser un algorithme de routage qui a pour rôle de trouver le meilleur chemin possible entre la source et le destinataire, ainsi que de garantir un certain niveau de qualité.

Il existe plusieurs types d'algorithmes, on cite ceux du :

- Routage fixe
- Routage adaptatif

4. La qualité de service (QoS)

Après avoir vu les principaux facteurs qui définissent un réseau, il est maintenant nécessaire d'assurer une qualité de service en gérant la bande passante et en définissant des priorités.

4.1 Définition

La QoS (Quality of Service) **[I.3]** est un ensemble de fonctionnalités qui permettent de différencier les flux applicatifs et d'apporter une priorité aux applications critiques. Elle permet de faire fonctionner des applications multimédia (voix-vidéo) sur un réseau initialement conçu pour les applications informatiques.

Applications critiques	5%
Temps réel (voix-vidéo)	65%
Applications courantes	30%

Tableau I-1 Répartition de la bande passante

Elle optimise l'utilisation de la bande passante sur liens d'interconnexion, et améliore sensiblement le fonctionnement de l'infrastructure par :

- la gestion d'une bande passante dédiée aux applications critiques et aux utilisateurs à servir en priorité.
- le contrôle de la gigue et du temps de latence dans les nœuds (routeur) du réseau
- la réduction et le contrôle de congestions dans le réseau.
- la régulation du trafic.
- la définition des priorités au sein du réseau.

D'une manière générale, la QoS est basée sur un traitement prioritaire des flux applicatifs selon leur nature. Elle raisonne ainsi sur un pourcentage de la bande passante attribuée à chaque protocole ou famille de protocoles.

La qualité de service est implémentée sur les différents équipements de l'architecture d'un réseau (routeur, switch, etc.) comme le montre la figure I-7.

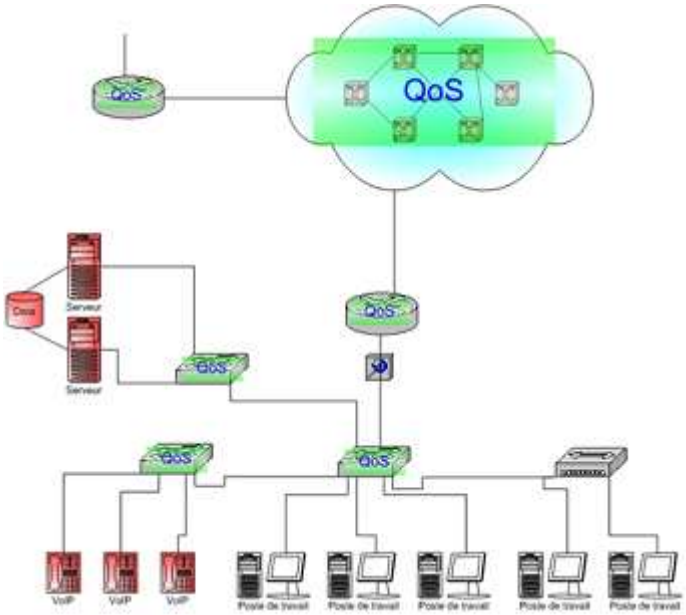


Figure I-7 : Implémentation de la QoS

4.2 Gestion de la QoS

Il existe 3 approches pour gérer la QoS :

- traitement du paquet IP suivant la priorité indiquée dans son en-tête.
- affectation des priorités et des classes des services. Ces valeurs sont transportées dans les paquets IP. Le formatage se fait en entrée du réseau et le traitement au sein de ce dernier.
- réservation des ressources nécessaires à la communication, tout au long du chemin qu'emprunteront les paquets. Ensuite tous les paquets de cette communication suivront la politique de qualité de service mise en place lors de la réservation

La gestion de la QoS suit plusieurs étapes :

1. La file d'attente
   - Traite en priorité tel ou tel paquet en cas de congestion ;
   - Rejette en priorité tel ou tel paquet en cas de saturation ;

2. Le classificateur
   - Affecte une priorité ou/et une classe de service au paquet en fonction de ses paramètres.

3. L'ordonnanceur
   - Mesure et analyse le flux (metering)
   - Ensuite compare avec la QoS demandée
   - Puis applique la qualité de service en fonction de la priorité ou de la classe à laquelle appartient le paquet. Il peut alors :
     - Changer le marquage du paquet
     - Réguler le trafic
     - Écrêter les pics de trafic

5. Types de mesures [I.4]

Il existe deux façons pour mesurer les performances d'un réseau : une active et l'autre passive.

5.1 Mesures passives

5.1.1 Principe

Le principe des mesures passives est l'observation du trafic et puis l'étude de ses propriétés en un ou plusieurs points d'un réseau. L'endroit idéal pour positionner des sondes de mesures passives est indéniablement dans les routeurs, car c'est par là que le trafic transite le plus. (cf. figure I.8)

L'avantage des mesures passives est qu'elles ne sont absolument pas intrusives, du point de vue réseau, et ne changent rien à son état. En revanche, il est très difficile de déterminer le service qui pourra être offert à un client, en fonction des informations obtenues par métrologie passive, car cette approche ne permet pas de mesurer les limites des performances d'un réseau.

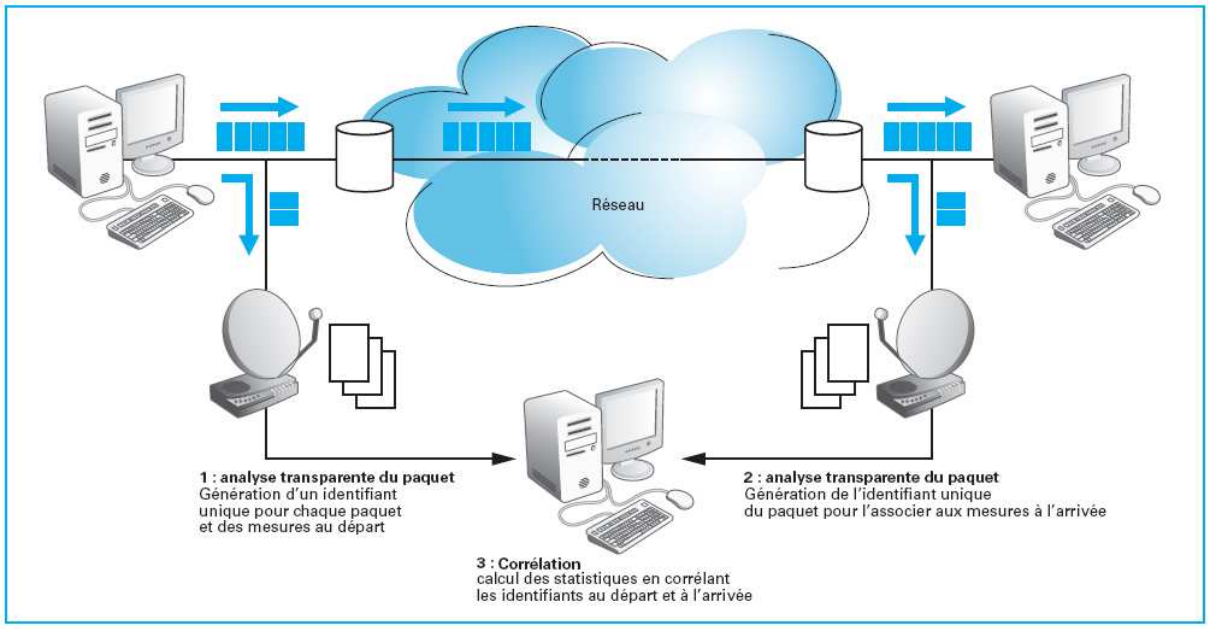


Figure I.8 : Architecture passive

5.1.2 Les problématiques

La principale contrainte qui se pose pour l'installation de sondes de mesure est due au fait que le réseau, dont nous souhaitons analyser le trafic, est presque toujours un réseau opérationnel et que malgré la présence de la sonde, celui-ci doit continuer à fonctionner sans aucune dégradation du service qu'il offre.

5.1.3 Exemples d'outils

De nombreux outils sophistiqués ont été développés. Les infrastructures de mesures passives ont en général recours à des équipements dédiés. Les plus connus sont les cartes DAG **[I.5]**.

Des boîtiers spécifiques sont également disponibles, par exemple les solutions proposées par Ipanema **[I.6]** ou par QoSMetrics **[I.7]**. Les outils de mesure passive sont en général limités par :

- leur capacité à analyser les paquets passant sur un lien à sa vitesse ;
- leur capacité de stockage des données prélevées ;

Parmi les exemples d'outils utilisant la métrologie passive on peut aussi citer le standard SNMP (voir chapitre II)**.**

5.2 Mesures actives

5.2.1 Principe

Le principe des mesures actives consiste à générer du trafic dans le réseau à étudier, puis de constater ses effets sur le comportement global de ce dernier : taux de perte, délai, RTT, etc. (cf. figure I-9)

Cette approche possède l'avantage de prendre un positionnement orienté utilisateur. Les mesures actives restent le seul moyen, pour un utilisateur, de mesurer les paramètres du service dont il pourra bénéficier.

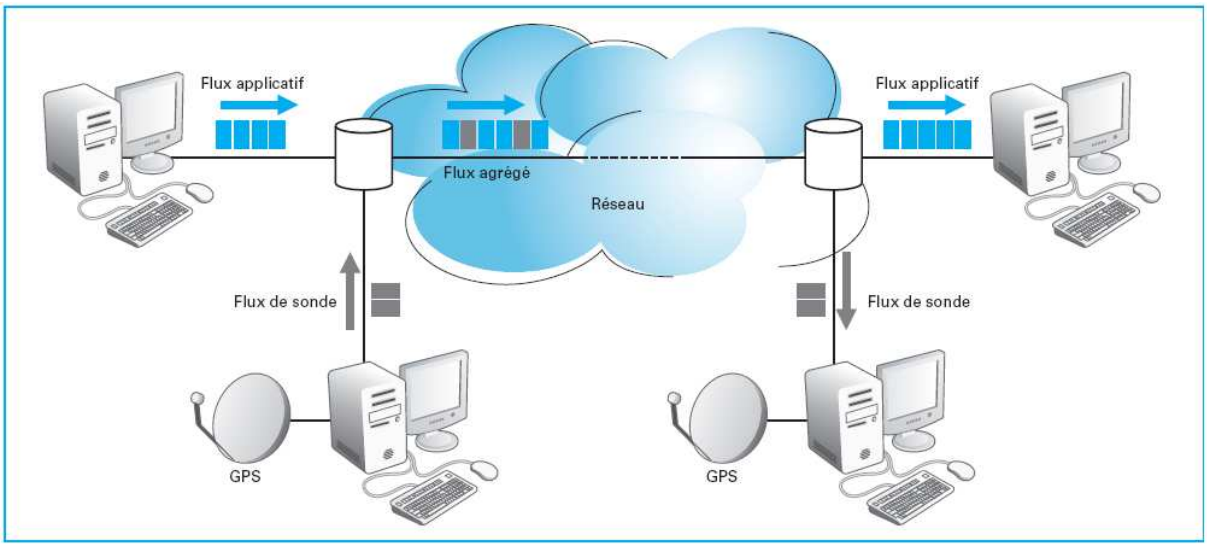


Figure I-9 : Architecture active

5.2.2 Les problématiques

L'un des inconvénients majeurs pour un réseau, dans le cas des mesures actives, est la perturbation introduite par le trafic ajouté. Cette dernière peut faire évoluer l'état de ce réseau et ainsi fausser la mesure.

En effet, le résultat de ces mesures donne une information sur l'état du réseau transportant à la fois :

- les données normales des utilisateurs et de signalisation du plan de contrôle du réseau,
- l'ensemble des paquets sondes.

Or, on souhaiterait avoir une information qui correspond au trafic normal uniquement, sans les paquets sondes qui ont forcément un impact sur les performances du réseau.
Il faut donc, soit être capable d'estimer cet impact et le rendre minimal, voir nul.

C'est cette dernière proposition, simple à priori, qui suscite le plus d'efforts de recherche. On parle de trafic de mesure non intrusif.

Ainsi, de nombreux travaux menés actuellement abordent ce problème en essayant de trouver les profils de trafic de mesure qui minimisent les effets du trafic supplémentaire sur l'état du réseau. C'est, par exemple, le travail en cours au sein du groupe IPPM de l'IETF **[I.8] [I.9] [I.10][I.11]**.

5.2.3 Exemple d'outils

Les mesures actives simples restent, tout de même, monnaie courante sur Internet pour lequel de nombreux outils de test, validation et/ou mesure sont disponibles. Parmi eux, on peut citer les célèbres requêtes ICMP : *ping* et *traceroute*.

Ping permet de vérifier qu'un chemin est valide entre deux stations et de mesurer certains paramètres comme le RTT ou le taux de perte.

Traceroute fait apparaître l'ensemble des routeurs traversés par les paquets émis jusqu'à leur destination et donne une indication sur les temps de passage sur chacun de ces nœuds.

Conclusion

Nous avons présenté les notions fondamentales relatives à la métrologie des réseaux. Cette dernière propose deux approches : l'une dite active et l'autre passive.

Le but de ces mesures est la supervision d'un réseau et l'amélioration de sa gestion en aidant à la définition, à la mise en œuvre et au maintien d'une qualité de service adéquate.

Dans la suite de ce rapport, on va s'intéresser à la métrologie passive via le protocole SNMP. Elle constituera un pré requis nécessaire à la mise en place d'une plateforme de supervision pour les équipements d'une infrastructure de réseau donnée.

CHAPITRE II

Mesures passives via le protocole SNMP

Introduction

Les outils de surveillance d'un réseau sont indispensables aux administrateurs afin de détecter des anomalies de fonctionnement, qu'elles soient dues à un équipement défectueux ou à une attaque. Ils offrent une vision synthétique du réseau et de son état qui permet, outre de générer une alarme en cas de problème, de pouvoir mieux connaître l'utilisation du réseau afin d'en prévoir les évolutions. **[II.1]**

Il est de ce fait souhaitable d'appuyer autant que possible l'administration des réseaux sur des standards comme le protocole SNMP (Simple Network Management Protocol) **[II.2][II.3][II.4][II.5]** qui est actuellement la technologie de base permettant d'administrer un réseau IP, et plus généralement les réseaux informatiques (systèmes, applications, etc).

Même si ce n'est pas la technologie permettant de résoudre tous les problèmes, il convient de la mettre en œuvre dès que l'on a affaire à un réseau local dont on veut apprécier le comportement.

Les réseaux de télécommunications (opérateurs) s'appuient quant à eux sur CMIP (Common Management Information Protocol) et sur l'architecture TNM (Telecommunications Network Management). CISCO utilisent un outil propre à eux qui est désigné par Netflow **[II.6]** pour la supervision de leurs équipements. **[II.7]**

Dans ce chapitre, on présentera le protocole SNMP d'une manière générale en abordant son principe de fonctionnement, ses évolutions et sa mise en œuvre sur des systèmes d'exploitation différents.

1. SNMP et les mesures passives :

Les mesures passives se basent sur l'observation non intrusive du comportement d'un réseau. Elles se basent sur la mise en place d'un certain nombre de sondes au niveau de cette infrastructure. Le protocole SNMP est utilisé dans ce contexte, dans le but d'interroger les éléments actifs d'un réseau via les sondes qui y sont déployées.

2. Présentation

SNMP (Simple Network Management Protocol) est le protocole standardisé pour l'administration du réseau Internet et il est également très largement déployé dans les réseaux d'entreprise.

SNMP est un protocole bâti au dessus d'UDP/IP :

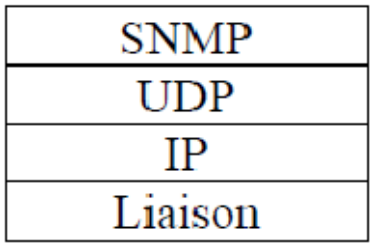


Figure II-1 : SNMP et la pile de protocoles IP

L'utilisation du protocole UDP est justifiée pour sa simplicité, sa rapidité et sa concision (8 octets d'entête contre 20 pour TCP), ce qui permet de remonter très rapidement des alarmes vers un manager. Malheureusement, c'est un protocole de transport en mode non connecté et non fiable, ce qui signifie que l'on pourra perdre des messages SNMP. **[II.8]**

3. Principe

3.1 Un Agent sur chaque équipement

Chaque équipement que l'on voudra "manager" à distance devra disposer d'un Agent SNMP. Cet Agent est un serveur, c'est à dire qu'il reste à l'écoute d'un port particulier : le port UDP 161.

La principale fonction de cet Agent est de rester à l'écoute des éventuelles requêtes que l'administrateur lui enverra. Lorsqu'il recevra une requête, il y répondra, s'il y est autorisé. Plus exactement, il répondra si la requête est émise par une entité autorisée. Autrement dit, cet Agent est là pour écouter des requêtes et y répondre.

L'Agent devra éventuellement pouvoir agir sur l'environnement local, si l'administrateur souhaite modifier un paramètre.

Par ailleurs, l'Agent SNMP pourra éventuellement émettre des alertes de sa propre initiative, s'il a été configuré pour ça. Par exemple, il pourra émettre une alerte si le débit sur une interface réseau atteint une valeur considérée par l'administrateur comme critique. Il peut y avoir une multitude d'alertes possibles, suivant la complexité de l'Agent : la température du processeur, le taux d'occupation des disques durs, le taux d'occupation CPU...

On pourra trouver des Agents SNMP sur des hôtes (ordinateurs) mais aussi sur des routeurs, des ponts, des switches...

3.2 Un "Manager" sur la station d'administration

L'administrateur, sur sa machine d'administration, dispose d'un outil dit "Manager". C'est avant tout un client, dans la mesure où c'est lui qui envoie les requêtes aux divers Agents SNMP du réseau. Il devra aussi disposer d'une fonction serveur, car il doit rester à l'écoute des alertes que les divers équipements sont susceptibles d'émettre à tout moment.

Sur la figure II-2, tous les équipements disposent d'un Agent SNMP, la station d'administration dispose du "Manager"

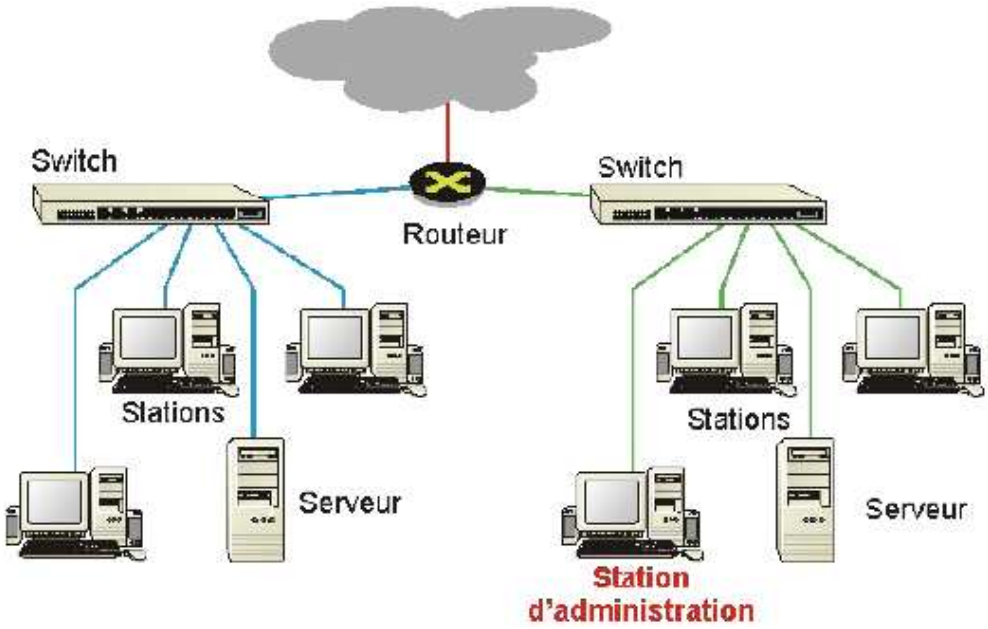


Figure II-2 : Principe d'utilisation du protocole SNMP

Dans un tel cas, l'administrateur peut, en théorie, observer le comportement de la totalité de son réseau depuis sa station d'administration. C'est vrai pour un LAN, c'est aussi vrai pour un

WAN (entendez par là que l'équipement à administrer peut se trouver à des centaines de kilomètres).

Les "boîtes noires" (routeurs, switches...) sont équipées éventuellement d'un agent SNMP (c'est une question de prix).

Pour les serveurs et les stations il existe probablement un logiciel à installer, quelque soit le système. Pour Linux, c'est "NET-SNMP" (ou "UCD-SNMP"), Windows XP professionnel, Windows 2000 "pro" et "server" permettent d'installer un agent SNMP.

Le Manager dispose d'un serveur qui reste à l'écoute, sur le port UDP 162, des éventuels signaux d'alarme. **[II.9]**

Donc, SNMP s'appuie essentiellement sur UDP et utilise aussi un nombre très restreint de commandes.

Les commandes sont résumées dans le tableau II-1 :

Commandes	Description
get-request	Le Manager SNMP demande une information à un agent SNMP
get-nextrequest	Le Manager SNMP demande l'information suivante à l'agent SNMP
set-request	Le Manager SNMP met à jour une information sur un agent SNMP
get-response	L'agent SNMP répond à un get-request ou a un set-request
trap	L'agent SNMP envoie une alarme au Manager

Tableau II-1 : Liste des requêtes SNMP

Ces commandes seront mieux explicitées un peu plus loin dans ce chapitre.

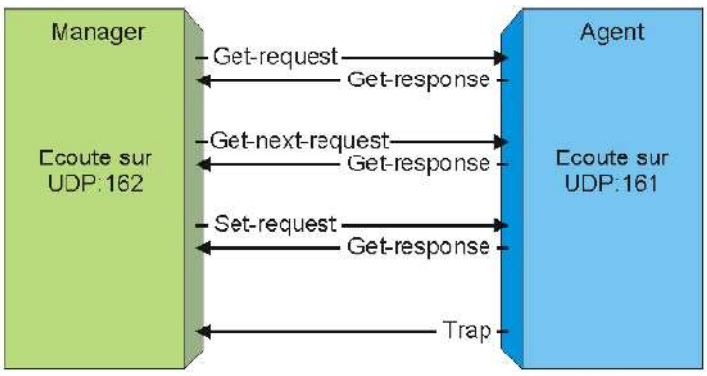


Figure II-3 : Schéma de communication Agent-Manager

4. Versions et évolutions du protocole SNMP

Actuellement, trois versions d'SNMP coexistent :

- **SNMP v1** est encore la plus utilisée en 2001, bien que ce soit la plus rudimentaire.

- Il y a eu plusieurs versions expérimentales de **SNMPv2**, qui devait remplacer SNMPv1, et en particulier lui apporter les fonctions de sécurité qui lui font défaut. Mais, faute d'un consensus au niveau des groupes de travail de l'IETF, c'est la version intermédiaire et expérimentale connue sous le nom de **SNMPv2C** qui est utilisée par la plupart des éditeurs supportant SNMPv2.

  La sécurité est encore quasiment nulle car elle reprend le modèle de SNMPv1, à base de « noms de communauté » (d'où le C de SNMPv2C). Malgré tout, elle comble des lacunes de la version 1, en particulier au niveau de la définition des objets, du traitement des notifications et du protocole lui-même (ajoutant une commande GETBULK de manière à minimiser les échanges réseau, particulièrement lourds dans le cas de récupération de tables avec les commandes GETNEXT de SNMPv1).
  Cette version n'a pas eu un important succès au niveau des déploiements sur les réseaux, faute de progrès conséquents par rapport à la version 1.

- **SNMPv3** apporte essentiellement des fonctions de sécurité et formalise de façon complète le modèle d'administration SNMP. C'est le standard cible qui reste encore à être éprouvé par le marché. **[II.10]**

Le tableau ci-dessous montre ces 3 versions avec leurs références respectives qui sont des RFC dans ce cas :

Version	Année	RFCs	Titre	Statut
v1	1990	1155	Structure and Identification of Management Information for TCP/IP-based Internets	standard
		1156	Management Information Base for network management of TCP/IP-based Internets	historique
		1157	Simple Network Management Protocol (SNMP)	historique
v2c (classic)	1993	1441	Introduction to version 2 of the Internet-standard Network Management Framework	historique, proposé comme standard
		1442	Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard proposé, remplacé par RFC-1902
		1443	Textual conventions for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard proposé, remplacé par RFC-1903
		1444	Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard proposé, remplacé par RFC-1904

		1445	Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)	Historique
		1446	Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2)	Historique
		1447	Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2)	Historique
		1448	Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard proposé, remplacé par RFC-1905
		1449	Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard proposé, remplacé par RFC-1906
		1450	Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard proposé, remplacé par RFC-1907
		1451	Manager-to-Manager Management Information Base	Historique
		1452	Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework	Standard proposé, remplacé par RFC-1908
v2	1996	1901	Introduction to Community-based SNMPv2	Historique, proposé comme expérimental
		1902	Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard, remplacé par RFC-2578
		1903	Textual conventions for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard, remplacé par RFC-2579
		1904	Conformance Statements for version 2 of the Simple Network Management Protocol (SNMPv2)	Standard, remplacé par RFC-2580
		1905	Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2) (3416)	Standard, remplacé par RFC-3416
		1906	Transport Mappings for version 2 of the Simple Network Management Protocol (SNMPv2) (3417)	Standard, remplacé par RFC-3417
		1907	Management Information Base for version 2 of the Simple Network Management Protocol (SNMPv2) (3418)	Standard, remplacé par RFC-3418
		1908	Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework (2576)	Standard, remplacé par RFC-2576
v3	1999	2571	An Architecture for Describing SNMP Management Frameworks	Standard, remplacé par RFC-3411
		2572	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	Standard, remplacé par RFC-3412
		2573	SNMP Applications	Standard, remplacé par RFC-3413

		2574	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	Standard, remplacé par RFC-3414
		2575	View-based Acces Control Model (VACM) for the Simple Network Management Protocol (SNMP)	Standard, remplacé par RFC-3415
v2 et v 3	2000	2576	Coexistence between Version 1, 2 and 3 of the Internet-standard Network Management Framework	Standard proposé
	2002	3411	An Architecture for Describing Simple Network Management Protocol (SNMP) Frameworks	Standard
		3412	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	Standard
		3413	Simple Network Management Protocol Applications	Standard
		3414	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	Standard
		3415	View-based Acces Control Model (VACM) for the Simple Network Management Protocol (SNMP)	Standard
		3416	Version 2 of Protocol Operations for the Simple Network Management Protocol (SNMP)	Standard
		3417	Transport Mappings for version 2 of the Simple Network Management Protocol (SNMP)	Standard
		3418	Management Information Base for version 2 of the Simple Network Management Protocol (SNMP)	Standard

Tableau II-2 : Evolutions du protocole SNMP

5. La communauté SNMP

La communauté SNMP (SNMP Community) a deux sens :

- On appelle communauté l'association Agent(s) – Manager. Les appareils pouvant êtres supervisés (serveurs, éléments actifs de réseaux…) pourraient constituer une communauté.

  A noter qu'un agent peut appartenir à plusieurs communautés, puisque, selon son paramétrage, il peut connaitre plusieurs Managers.

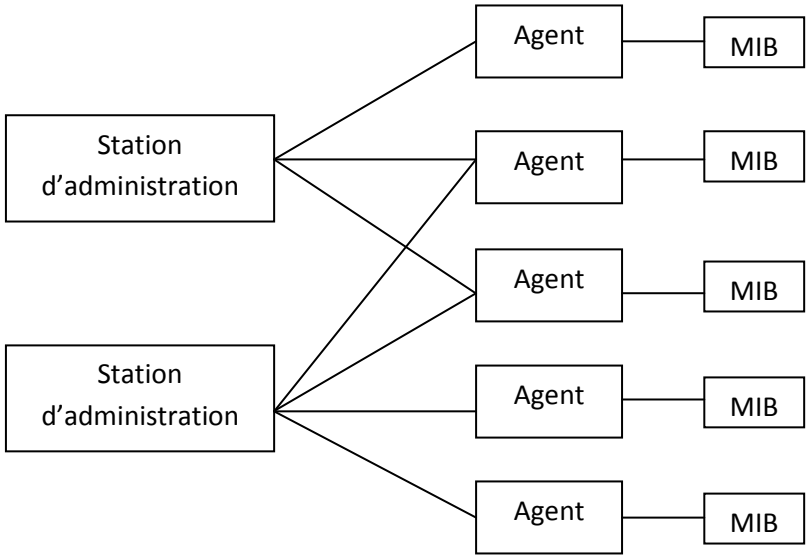


Figure II-4 : Communauté SNMP

Les noms de communauté peuvent également être associés à des niveaux d'accès aux MIBs différentes (droits de lecture seule RO, ou de lecture/écriture RW).

- Ensuite, la chaine de caractères correspondant au nom de la communauté sera utilisée, dans la version 1 du protocole SNMP, comme mot de passe pour l'accès en lecture ou en écriture à la MIB.

Cela constitue une faiblesse notoire en matière de sécurité : le nom de communauté est transmis en clair dans les messages SNMP et toute modification de la MIB correspond de fait à une configuration d'un appareil (suppression d'une entré dans une table de routage, ou fermeture administrative d'un port de commutateur) pouvant entrainer un disfonctionnement du réseau. N'importe quel analyseur de protocole (sniffer) permet de découvrir aisément les noms de communautés, en lecture comme en écriture.

A noter que bien des éléments actifs de réseau fonctionnent avec les noms de communauté par défaut (*public* pour lecture seule et *private* pour lecture/écriture). **[I.3]**

6. Structure des messages SNMP

Comme SNMP version 1 (v1) propose cinq messages :

- Get_Request
- Get_Next_Request
- Get_Response

- Set_Request
- Trap

Le format générique des messages SNMP est le suivant :

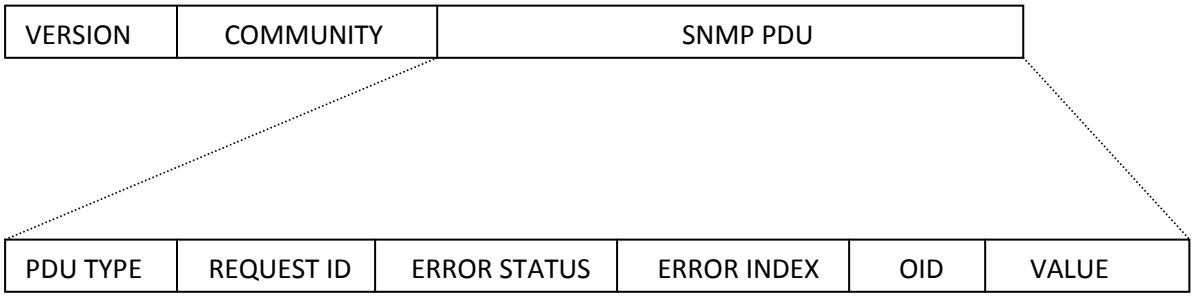


Figure II-5 : Format générique des messages SNMP

On présentera notamment les champs les plus importants à savoir :

Version : Indique la version du protocole SNMP utilisé :

- 0 : SNMP v1
- 2 : SNMP v2
- 3 : SNMP v3

Community : Nom de communauté de lecture seule (RO) ou de lecture/écriture (RW).

PDU TYPE : Description du type de message SNMP.

- 0 : Get_Request
- 1 : Get_Next_Request
- 2 : Get_Response
- 3 : Set_Request
- 4 : Trap

Request ID : Identifiant du message SNMP, utilisé par le manager pour la vérification de cohérence entre requête et réponse.

Error Status / Error Index : Renvoie les éventuelles erreurs :

- NoAcces : Accès à la MIB non permis.

- WrongLength : erreur de longueur de la requête.
- WrongValue : valeur de la donnée erronée.
- WrongType : type de donnée erroné.
- WrongEncoding : erreur d'encodage.
- NoCreation : objet non créé.
- ReadOnly : impossibilité de modifier la variable.
- NoWritable : pas de permission d'écriture.
- AuthorisationError : erreur d'autorisation.

OID : Indicateur de variable concernée par la requête (ou la réponse).

Value : Information renvoyée par l'Agent ou destinée à la mise à jour de la MIB, émise par le Manager. **[I.3]**

7. Les types de requêtes SNMP

7.1 Le message Get_Request

Le message Get_Request constitue une requête émise par le Manager à l'attention de l'Agent pour obtenir une information de la base de données (récupération d'un attribut d'une variable).

SNMP v1 limite les requêtes à une seule variable. Il est par exemple impossible au Manager de prendre connaissance, avec une seule requête (dans la limite des 484 octets du champ de données du datagramme UDP), des statistiques concernant l'ensemble des ports d'un commutateur Ethernet. Le Manager émettra autant de Get_Request que nécessaire.

A noter que

- La requête Get_Request ne permet que d'interroger l'Agent sur une variable connue (il est donc impossible de connaitre toutes les entrées de la table de routage d'un routeur).
- Le nom de communauté RO (read only) est transmis dans la requête.
- Les services en mode non connecté du protocole UDP n'assurant pas au Manager la réception de sa requête par l'Agent concerné, il lui appartient de gérer les éventuelles

réémissions en cas de dépassement d'un temps donné (paramétrage d'un timer dans le Manager).

- La requête porte un identifiant (Request ID) recopié par l'Agent dans sa réponse, permettant l'association requête/réponse et la verification de la cohérence du dialogue par le Manager (risque d'une éventuelle duplication des messages et différenciation de la perte d'une requête ou d'une réponse).
- Toute réponse dont la taille est supérieure à 484 octets étant impossible, l'Agent retourne un code d'erreur (WrongLength).
- L'Agent retournera également un code d'erreur (NoAccess) en cas de discordance entre les noms des communautés. **[I.3]**

7.2 Le message Get_Next_Request

Il permet de balayer la MIB complète de l'Agent a priori inconnue du Manager (nombre d'entrées dans la table de routage d'un routeur, par exemple).

La requête Get_Next_Request est, dans un premier point, identique à la précédente. Elle d'interroger l'Agent et de récupérer la valeur (attribut) de la variable dont l'OID suit, dans l'arborescence, l'OID transporté dans la requête.

Exemple

Une requête du Manager Get_Request avec l'OID 1.3.6.1.2.1.1.1 (sysDescr) obtiendra une réponse de l'Agent avec l'OID 1.3.6.1.2.1.1.2 (sysObjectID).

L'association de deux messages SNMP Get_Next_Request et Get_Response permet ainsi d'interroger l'Agent sur sa MIB complète. Ce mécanisme est notamment utilisé pour la découverte automatique des éléments actifs du réseau par un Manager dans ses opérations de configuration. **[I.3]**

7.3 Le message Get_Response

C'est le message envoyé par l'Agent au Manager à la suite d'une requête. Il est impossible à l'Agent d'émettre spontanément un message Get_Response sans avoir été préalablement sollicité.

L'information renvoyée peut se présenter sous la forme :

- NetworkAddress : adresse MAC.
- IpAdress : adresse IP.
- Counter : information cumulée.
- Gauge : indicateur statistique.
- TimeTicks : mesure de temps.
- OctetString : chaine de caractères.

La réponse peut être nulle (exemple de non concordance des noms de communautés) mais transportera toujours l'identifiant de la requête Request ID (voir Get_Request) et un code d'erreur (Error Status). **[I.3]**

7.4 Le message Set_Request

Le message Set_Request permet, dans certaines conditions, de modifier la valeur d'une variable.

Ce message remplace simplement l'attribut concerné par l'OID. Il est en effet impossible de créer une variable, comme une entré dans la table de routage d'un routeur.

Un message Set_Request est toujours suivi d'un message Get_Response, avec les mêmes valeurs de Request ID, pour le contrôle de la cohérence requête/réponse par le Manager.

Une modification infructueuse de la MIB est notifiée au Manager par le code d'erreur correspondant, transporté par le message de réponse (NoAccess, WrongValue….).

A noter que :

- Le nom de communauté transporté dans le message doit correspondre au nom de communauté lecture/écriture paramétré par l'Agent.
- La variable concernée doit pouvoir être modifiée ce qui n'est pas, par exemple, le cas d'une adresse MAC.
- La concurrence d'accès entre Managers n'est pas gérée. **[I.3]**

7.5 Le message Trap

Trap est le seul message SNMP pouvant être envoyé spontanément par l'Agent à un Manager (ou plusieurs Managers suivant sa configuration). Cela se produit en cas d'incident grave. La RFC 1157 de SNMP v1 prévoit les cas suivants :

- ColdStart_Trap (0) : émis en cas de redémarrage complet de la machine (cas d'une remise sous tension après coupure ou par intervention d'un opérateur).
- WarmStart_Trap (1) : signifie qu'un des services de la machine a été réinitialisé, sans que l'Agent lui-même ait été affecté.
- LinkDown_Trap (2) : indique la coupure d'une interface d'un routeur (coupure liaison inter-routeur) ou d'un commutateur Ethernet (incident sur un port).
- LinkUp_Trap (3) : signifie le routeur en mode normal après un LinkDown_Trap (rétablissement de la liaison de données entre routeurs, réactivation du port Ethernet).
- AuthentificationFailure_Trap (4) : émis, par exemple, par un commutateur Ethernet donc le filtrage par adresse MAC a été activé sur l'un des ports sur lequel est reçue une trame Ethernet dont l'adresse MAC source est interdite.
- EgpNeighborLess_Trap (5) : émis par un routeur en cas de rupture de liaison avec un voisin EGP (problème de routage par exemple).
- EnterpriseSpecific_Trap (6) : indique un événement spécifique à l'équipement (autre que ceux référenciés ci-dessus), message Trap associé dans ce cas à une MIB privée.

Le message Trap porte un champ TimeStamp permettant son horodatage.

Se rappeler ici que le protocole SNMP utilise les services du protocole UDP, en mode non connecté, et que les messages Trap peuvent ne jamais arriver au Manager (mauvaise gestion d'une collision par le protocole CSMA-CD, destruction d'un datagramme IP par un routeur, problème de routage…). De plus, rien n'est prévu pour notifier à l'Agent la bonne réception d'un message Trap par le Manager.

C'est une faiblesse grave de l'architecture SNMP qui peut s'illustrer de la manière suivante :

- Un détecteur d'incendie est installé dans un local technique ainsi qu'une électrovanne sur un extincteur, ces deux équipements étant doté d'un Agent SNMP adapté.
- Le Manager est la centrale d'alarme dans la loge du gardien.

- Une alarme d'incendie dans le local technique peut ne jamais arriver à la loge : le gardien n'est pas informé d'un incident grave.
- La commande d'ouverture de la vanne de l'extincteur peut ne jamais arriver : le gardien croit que l'incendie est maitrisé des suites de son action curative. **[I.3]**

8. Structure de la SMI

La structure SMI décrit les règles de description de l'information et permet d'identifier de façon unique un objet de la MIB géré par un agent SNMP. Chaque objet possède donc un identificateur unique ou OID (*Object ID*).

SMI s'intéresse aussi à la représentation des données (et leur type) pour chaque objet de la MIB. Un objet de la MIB est déclaré et défini en langage ASN.1 (*Abstract Syntax Notation 1* : langage de représentation de donnée).
SNMP n'utilise qu'une petite partie du langage ASN.1. Au niveau des types, seuls quelques uns sont utilisés comme :

- INTEGER : valeur entière sur 32 bits en complément à 2.
- OCTET STRING : chaîne de caractères.
- IpAddress : adresse IP.
- PhysAddress : adresse MAC (6 octets pour un réseau de type Ethernet).
- Counter : entier de 32 bits non signé qui s'accroît de 0 à (2exp32 –1) puis revient à 0.
- TimeTicks : compteur de temps sur 32 bits non signé en 1/100 de s. **[II.8]**

9. La MIB

9.1 Généralités [I.3]

La MIB (Management Information Base) est structurée sous une forme arborescente et les OIDs (Object IDentifier) permettent de parcourir la MIB jusqu'à atteindre la variable souhaitée pour rn lire ou modifier les attributs.

L'arborescence de la MIB standard est définie comme suit : 1.3.6.1.2.1

Ainsi, tout OID représentant une variable de la MIB standard débute par cette chaine 1.3.6.1.2.1 pouvant être interprété comme *iso.org.dod.internet.management.mib*.

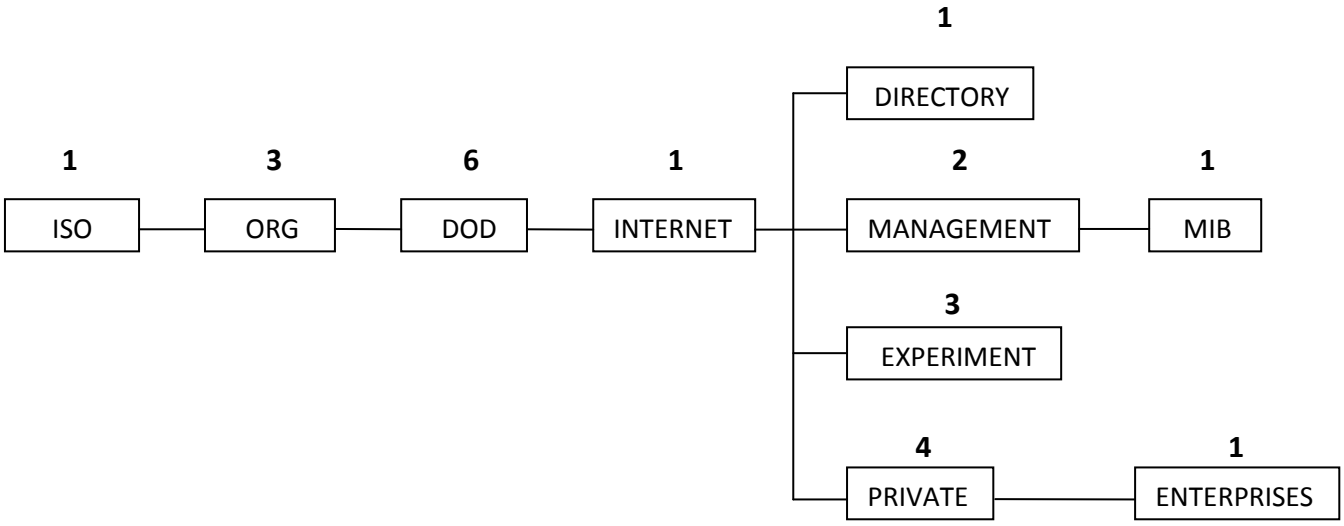


Figure II-6 - Arborescence de la MIB standard

A noter que les MIBs privées, développées spécifiquement par les constructeurs pour leurs appareils (quels qu'ils soient) ont comme préfixe 1.3.6.1.4.1 pour *iso.org.dod.internet.private.enterprises* **[II.11].**

Tout élément actif du réseau, même s'il intègre la MIB privée du manufacturier, doit supporter la MIB standard définie par l'IETF (mandatory implementation).

La MIB I dite standard a été très vite remplacée par la MIB II, un peu plus riche.

Tous les Managers, tous les Agents, tous les navigateurs de MIBs supportent les deux versions de la MIB standard :

- MIB I : RFC 1156 (mai 1990).
- MIB II : RFC 1213 (mars 1991).

9.2 La MIB I

Les variables gérées par la MIB I concernent principalement les routeurs. Il n'existait pas à cette époque d'autre appareil. Les standards IEEE 802.3 10base-T venaient d'être publiés, les hubs Ethernet du marché ne disposaient d'aucune ressource permettant leur supervision.

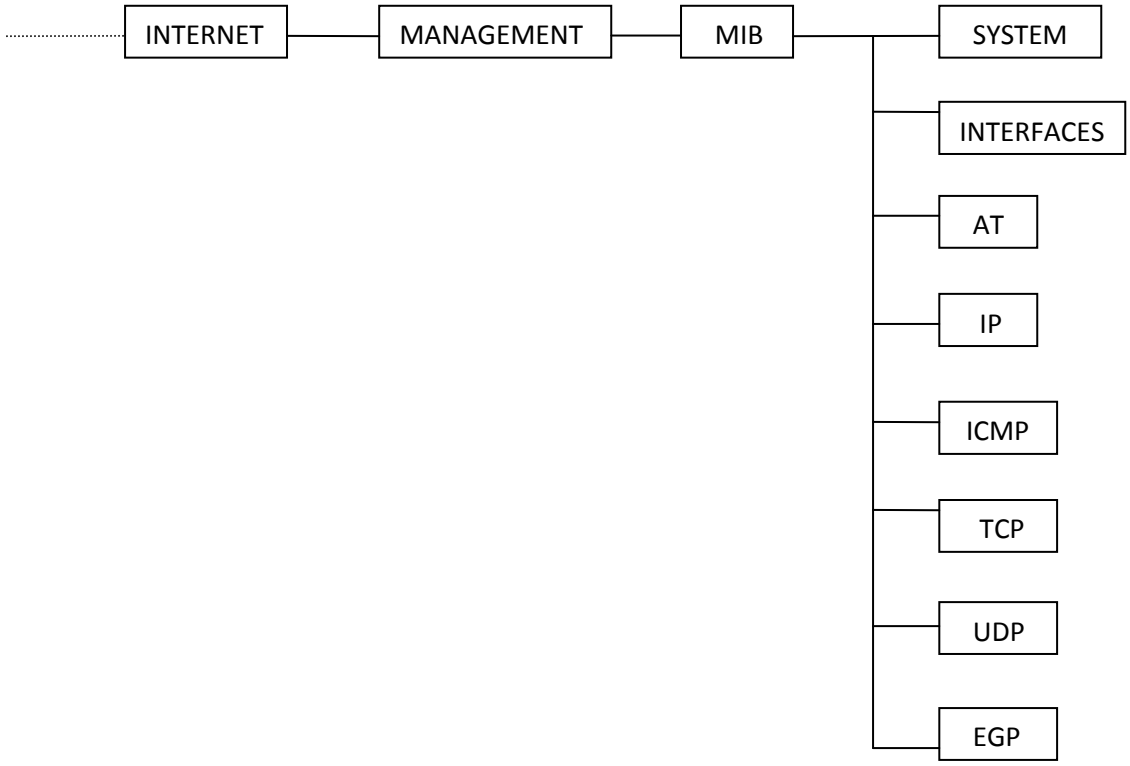


Figure II-7 : Arborescence de la MIB I

Les variables de la MIB I sont les suivantes :

1- **System Group :** Description formelle de l'entité (au format ASCII), avec identification claire de l'appareil (et éventuellement la version du system d'exploitation), OID correspondant à la MIB privée, historique de mise sous tension.

2- **Interfaces Group :** Nombre d'interfaces dont l'appareil est équipé, état administratif (validé ou bloqué), état opérationnel (actif ou inactif), type (Ethernet, Token-Ring, série…), compteur de trafic en entrée et en sortie au niveau de la couche Interface Réseau), cumul des erreurs, des collisions, des trames de broadcast…..

3- **Address Translation Group :** Corrélation entre adresses MAC et adresses IP (équivalence de la table ARP).

4- **IP Group :** Statistiques concernant le protocole IP, adresses IP affectées aux interfaces de la machine, table de routage (dans le cas d'un routeur IP).

5- **ICMP Group :** Statistiques concernant le protocole ICMP (nombre de messages échangés, de datagrammes détruits par les routeurs des suites de fragmentation…).

6- **TCP Group :** Indique les paramètres liés au protocole TCP : nombre de sessions actives supportées, de sessions ouvertes, de sessions anormalement rompues, de datagrammes TCP réémis des suites d'erreurs de transmission (checksum erronés)…

7- **UDP Group :** Indique les paramètres liés au protocole UDP : nombre de datagrammes UDP échangés, erreurs de checksum d'en-tête…

8- **EGP Group :** Obligatoire pour tous les systèmes qui implémentent le protocole de routage inter systèmes autonomes EGP (Exterior Gateway Protocol), donne les informations quant aux échanges protocolaires. **[I.3]**

9.3 La MIB II

Qui dispose en plus du groupe SNMP. Par ailleurs, le group System a été enrichi de variables supplémentaires. La MIB II est couramment implémentée, en lieu eu place de la MIB I.

A noter que si la MIB I est dédiée aux routeurs, la MIB II s'est adaptée aux autres éléments actifs du réseau. Les attributs de la variable sysServices, définis par la RFC 1213 sont les suivants :

- 1 : répéteur, hub Ethernet.
- 2 : pont, commutateur Ethernet, point d'accès (borne) Wi-Fi.
- 3 : routeur.
- 4 : host (station de travail, serveur).
- 7 : application (relais de messagerie).

Les variables sysContact, sysName et sysLocation peuvent être utilisées efficacement dans la notification d'alarme (voir le message Get_Response. Le traitement des alarmes et des événements). **[I.3]**

9.4 Les MIBs privées

9.4.1 Présentation

Rien, dans la MIB I, ni dans la MIB II ne permet de connaitre les paramètres d'environnement d'un châssis fédérateur (cœur de commutation dans un réseau Ethernet) en terme de température, de capacité de buffer ou de charge CPU, éléments pourtant essentiels en matière de stabilité et de maintien de performances.

Par définition, la MIB privée (private MIB) est définie spécifiquement par un constructeur pour son équipement. L'organisation générale de la MIB reste la même (arborescence, identification des informations par des OIDs…), mais les variables et leurs attributs sont intimement liés à l'appareil concerné et à sa fonction propre. Les mécanismes SNMP de lecture et de modification de la MIB privée sont bien entendu les mêmes que pour les MIBs standards I et II.

Pour un bon fonctionnement de la plate-forme et une parfaite interopérabilité Agent-Manager, la MIB privée doit être connue du Manager (voir Intégration dans un Manager).

Pour rappel, les OIDs des MIBs standards sont toujours préfixées par 1.3.6.1.2.1 pour la chaine *iso.org.dod.internet.management.mib*. en revanche, les OIDs des MIBs privées sont préfixées par 1.3.6.1.4.1 représentant la chaîne *iso.org.dod.internet.private.enterprises.*

Chaque entreprise (constructeur) a déposé un numéro auprès de l'ISO, pour l'organisation de sa MIB. Les affectations officielles sont organisées comme suit :

- 0 – 4999.
- 5000 – 9999.
- 10000 – 14999.
- 15000 – 19999.

Pour exemple, CISCO dont le numéro d'entreprise est 9. En interrogeant la MIB privée d'une borne Wi-Fi (AP1130AG) avec l'OID 1.3.6.1.4.1.9.9.414.1.1.3.1.3.3, le Manager prendra connaissance de l'état de l'interface radio 802.11G (disabled). **[I.3]**

9.4.2 Intégration dans un Manager

Les constructeurs d'éléments actifs des réseaux (routeurs, commutateurs Ethernet, bornes Wi-Fi…), de serveurs, d'imprimantes connectées mais aussi d'onduleurs et de boîtiers de contrôle d'environnement, développent nécessairement des MIBs privées, dont les sources sont livrées avec les appareils (sur CD-Rom couramment) ou librement téléchargeables sur leur site Internet et bien souvent largement documentées (cas de CISCO Systems notamment).

Les Managers disposent dans la plupart des cas (exception faite des logiciels d'entrée de gamme livrés avec les éléments actifs grand public) de fonctionnalités d'import des MIBs privées. Le format de la MIB étant alors défini (codage ASN.1), on parle de compilation de MIB. L'Agent, considéré à première vue comme exotique, est désormais intégralement connu du Manager qui pourra le solliciter pour prendre connaissance d'une de ses variables et se trouvera à même d'interpréter un message TRAP et de traiter comme prévu cet événement (notification par mail, relais du TRAP vers un NMS).

La compilation de MIB peut également être nécessaire quand un constructeur, par ailleurs éditeur de la suite logicielle de supervision et d'administration, annonce un nouvel équipement à priori inconnu dans sa liste native de produits administrables. Là encore, le constructeur offre la possibilité de télécharger les MIBs privées, souvent associées à des familles de produits (routeurs série 18xx, borne Wi-Fi série 11xx). Les révisions majeures (évolution de la v1.2 à la v2.0) des logiciels intègrent les mises à jour des catalogues de produits (incorporation des MIBs privées correspondantes).

Dans la plupart des cas, les MIBs des constructeurs sont documentées pour une meilleure compréhension de l'ingénieur réseau. L'arborescence est visualisable (affichage des chaînes et des OIDs), ainsi que les attributs possibles des différentes variables. **[I.3]**

10. La sécurité – communauté SNMP

La version 1 du protocole SNMP, définie pour mémoire par la RFC 1157 (mai 1990), ne traite pas la sécurité de l'accès à la MIB de l'Agent autrement qu'en associant, à chaque requête de lecture ou ordre de modification d'une variable, le nom de la communauté.

Ce nom doit être paramétré dans l'Agent, pour les droits de lecture seule (RO) et de lecture/écriture (RW), et connu du Manager.

Le nom de communauté est transmis en clair dans les messages SNMP et peut être aisément découvert avec les analyseurs de protocoles du marché. Il est alors facile de connaître l'intégralité de la configuration d'un élément actif du réseau ou de modifier sa configuration avec des requêtes simples dont la sémantique est parfaitement connue. **[I.3]**

Exemple de la configuration SNMP (Agent) d'une borne Wi-Fi Cisco Systems AP1130AG (les noms de communautés sont soulignés)

```
snmp-server community lectureseule  ro

snmp-server community lecturecriture rw
```

11. Mise en œuvre de SNMP sur des systèmes d'exploitation

L'installation, sous Linux comme sous Windows ne pose guère de problèmes. Il faudra juste faire un peu attention à ne pas laisser l'accès à n'importe qui sur n'importe quoi.

11.1 Sous Microsoft (Windows)

11.1.1 Installation

L'installation de SNMP sur les plateformes Microsoft est simple du fait qu'elle requiert uniquement son activation à travers l'assistant d'ajout de fonctionnalités lié au system (cf. figure II-8).

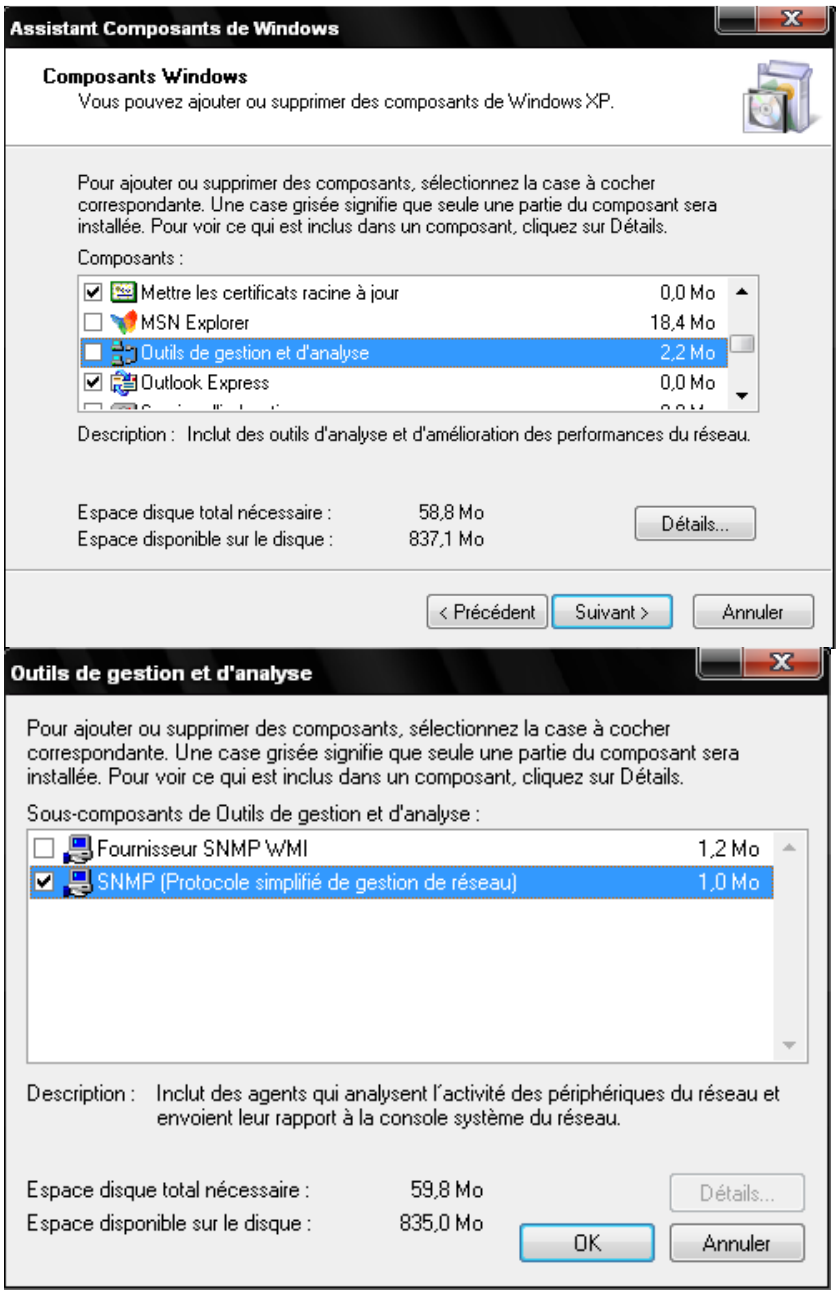


Figure II-8 : Installation de SNMP

11.1.2 Configuration

Cette étape est réalisée par l'édition du service SNMP sous le gestionnaire du système d'exploitation (cf. figure II-9)

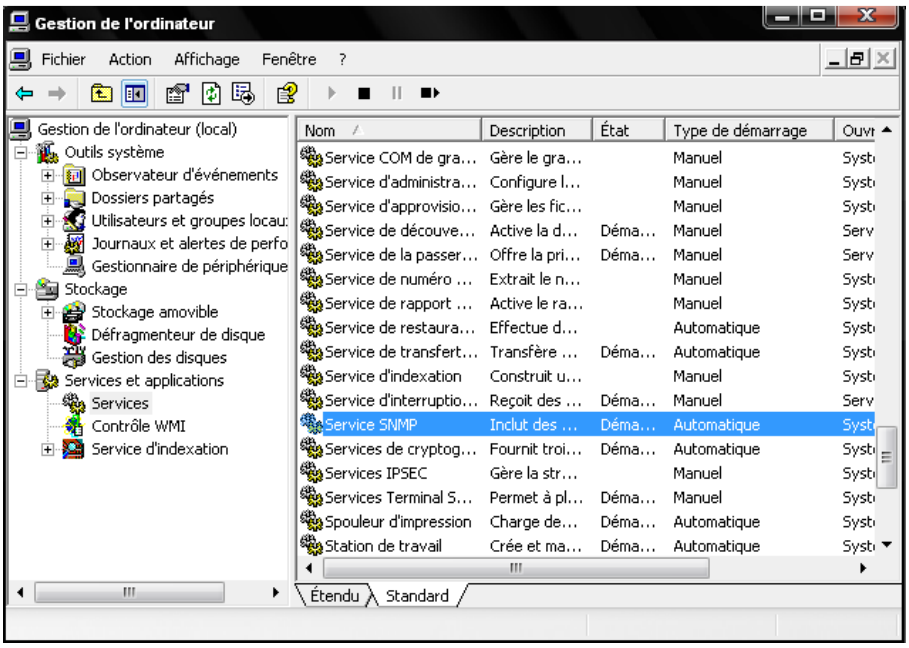


Figure II-9 : Recherche du service SNMP

Puis par la configuration de l'Agent SNMP (cf. figure II-10)

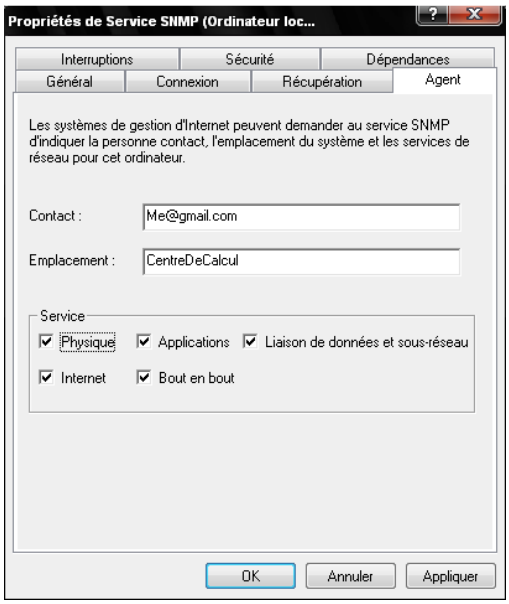


Figure II-10 : Configuration de l'Agent SNMP

Pour cela, on doit définir une communauté tout en spécifiant les droits attribués à cette dernière sans oublier de déclarer les machines (filtrage de niveau 3 : par adresse IP) ayant ces droits;

Exemple :

Une communauté ‘’public’’ ayant le droit de ‘’lecture seule’’ sur la MIB locale.

Une machine ‘’172.16.13.68’’ ayant le droit de consulter la MIB de cette machine Windows.

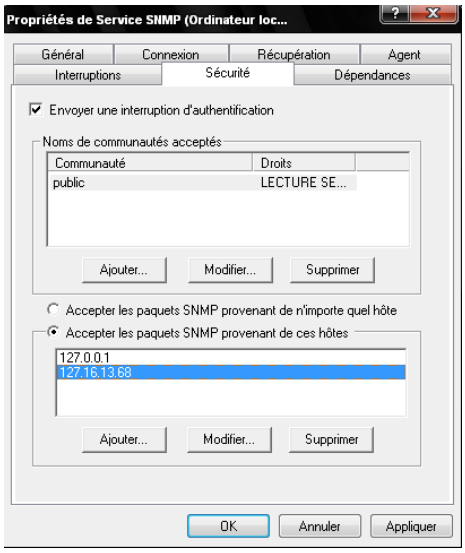


Figure II-11 : Exemple de configuration pour SNMP

11.2 Installation sous Linux

La mise en œuvre de SNMP sous Linux se fera en utilisant le standard de fait dans le logiciel libre : le package NET-SNMP.

Le projet NET-SNMP appelé anciennement UCD-SNMP a été historiquement développé par l’université américaine *Carnegie Mellon University* (CMU) puis amélioré et maintenu maintenant par l’université américaine *University of California Davis* (UCD).

NET-SNMP est en fait un ensemble d’outils et de fonctionnalités :

- Une API (*Application Programming Interface*) d’accès à SNMP.
- Un agent SNMP extensible.
- Des commandes en ligne pour interroger des agents SNMP.
- Des commandes en ligne pour gérer et générer des TRAPs SNMP.
- Une version de la commande UNIX netstat utilisant SNMP.
- Un browser de MIB SNMP (tkmib) écrit en Tk.

NET-SNMP est porté sur différents systèmes et en particulier sur :

- Linux (noyaux 2.4 à 1.3).
- HP-UX (10.20 à 9.01 et 11.0).
- Ultrix (4.5 à 4.2).

- Solaris (2.8 à 2.3) et SunOS (4.1.4 à 4.1.2).
- NetBSD (1.5alpha à 1.0).
- FreeBSD (4.1 à 2.2).
- Win32.
- …

NET-SNMP supporte SNMPv1, SNMPv2 et SNMPv3 que ce soit côté Agent SNMP comme du côté Manager SNMP via les commandes en ligne NET-SNMP.

11.2.1 Installation de NET-SNMP

Première méthode : à l'aide d'un package ‘’.tar.gz ‘’

L'installation de NET-SNMP est des plus traditionnelles sous Linux et se fait à l'aide des étapes décrites ci-dessous :

a) Décompression du paquetage :

```
# cd
# tar xvzf net-snmp-5.2.5.1.tar.gz
# ln -s net-snmp-5.2.5.1 net-snmp
```

b) Configuration. On choisira d'utiliser par défaut SNMPv2c :

```
# cd ~/net-snmp
# ./configure
```

c) Compilation:

```
# make
```

d) Installation :

```
# make install
```

Deuxième méthode : Par l'installation des paquets « deb »

Cette méthode requiert une distribution Linux compatible Debian, sur laquelle on ouvre un terminal pour la saisie de la commande suivante :

```
# apt-get install snmpd
```

A noter qu'il faut aussi installer le service SNMP par la commande :

```
# apt-get install snmp
```

Qui permettra par la suite de bénéficier des différentes requêtes SNMP traitée précédemment (voir structure des messages SNMP).

11.2.2 Configuration

Comme l'installation dans ce cas est faite sous une machine Ubuntu, l'agent SNMP NET-SNMP est l'exécutable snmpd sous */usr/sbin*. Il possède un fichier de configuration général s'appelant ''snmpd.conf'' à copier sous */etc/snmp*.

Si l'installation est faite avec la deuxième méthode ce fichier de configuration existe déjà dans */etc/snmp*. Sinon, il faudra le créer en ayant recourt à l'utilitaire **snmpconf** permet de créer le fichier snmpd.conf de façon interactive et conviviale sans en connaître exactement sa structure :

Création rapide du fichier snmpd.conf à la première utilisation :

```
# snmpconf –g basic_setup
```

Ou bien

Mode interactif :

```
# snmpconf
```

On pourra consulter l'aide en ligne sur la structure de ce fichier (# man *snmpd.conf*).

Exemple de configuration

Là aussi on choisira le même exemple que pour la configuration de l'Agent SNMP sous Windows.

Les champs les plus importants sont :

Déclaration d'une machine et d'un réseau IP ayant accès à l'agent SNMP, *readonly* et *readonly2* de communauté *public* :

```
#          sec.name      source          community
com2sec    readonly      localhost       public
com2sec    readonly2     172.16.13.86    public
```

Déclaration de groupes d'accès aux objets de la MIB de l'agent SNMP pour *readonly* et *readonly2* :

```
####
# Second, map the security names into group names:
#              sec.model          sec.name
group      MyRWGroup v1       readonly
group      MyRWGroup v2c      readonly
group      MyROGroup v1       readonly2
group      MyROGroup v2c      readonly2
```

Accès en lecture/écriture aux objets de la MIB de l'Agent pour l'accès *readonly* et lecture seulement pour l'accès *readonly2*

```
####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:

#            context          sec.model      sec.level     match     read    write     notif
access     MyROGroup ""        any            noauth        exact      all     none     none
access     MyRWGroup ""        any            noauth        exact      all      all      none
```

Autorisation de la génération de Traps SNMP en direction d'un manager SNMP de la machine *localhost* de communauté *public* :

```
####
# Traps and v2 traps enabled and sent to localhost
#
# command      host manager         community
    trapsink         localhost              tst
    trap2sink        localhost              tst
```

Renseignement de l'objet *sysLocation* de la branche *system* :

```
syslocation CCRSI-ENSP, Alger, Algérie
```

Renseignement de l'objet *sysContact* de la branche *system* :

```
syscontact Webmaster <Webmaster@enp.edu.dz>
```

Remarque

Une fois que l'édition du fichier de configuration terminée, il faudra par la suite relancer le service NET-SNMP en ayant recourt à la commande :

```
# /etc/init.d/snmpd restart
```

12. Quelques possibilités d'utilisation

Les principales commandes utiles notées globalement snmpxxx sont :

- snmpget : envoi d'une requête SNMP GET pour obtenir une information sur un objet de la MIB d'un agent SNMP distant.
- snmpset : envoi d'une requête SNMP SET pour mettre à jour la valeur d'un objet de la MIB d'un agent SNMP distant.

- snmpgetnext : envoi d'une requête SNMP GETNEXT et donne aussi la valeur de l'objet suivant de la MIB d'un agent SNMP distant si toutefois il en existe un.
- snmpwalk : cette commande fonctionne comme snmpgetnext mais permet de balayer complètement une branche de la MIB d'un agent SNMP distant.
- snmptranslate : permet de convertir un objet d'une MIB représenté sous sa forme décimale OID en sa forme symbolique et réciproquement.

Une fois l'agent SNMP lancé sur sa machine comme précédemment, il est ensuite possible de le tester avec les commandes snmpxxx.

Correspondance OID et nom symbolique :

```
# snmptranslate 1.3.6.1.2.1.1.3.0
SNMPv2-MIB::sysUpTime.0
```

Représentation sous forme graphique de la branche system de la MIB par analyse de l'ensemble des fichiers MIB sous */usr/share/snmp/mibs* :

```
# snmptranslate -Tp -IR system
+--system(1)
       |
       +-- -R-- String    sysDescr(1)
       |       Textual Convention: DisplayString
       |       Size: 0..255
      . . .
```

Récupération par SNMPv1 de la valeur courante de l'objet sysUpTime géré par l'agent SNMP de la machine localhost :

```
# snmpget -v 1 -c public localhost system.sysUpTime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (12908) 0:02:09.08
```

Récupération de la valeur courante de l'objet sysUpTime avec SNMPv2c :

```
# snmpget -v 2c -c public localhost system.sysUpTime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (13966) 0:02:19.66
```

Récupération de la valeur courante des objets sysLocation et sysContact (renseignés dans le fichier snmpd.conf) :

```
# snmpget -v 1 -c public localhost system.sysLocation.0
SNMPv2-MIB::sysLocation.0 = STRING: CCRSI-ENSP, Alger, Algérie
# snmpget -v 1 -c public localhost system.sysContact.0
SNMPv2-MIB::sysContact.0 = STRING: Webmaster <Webmaster@enp.edu.dz>
```

Parcours de la branche system de la MIB de l'agent SNMP :

```
# snmpwalk -v 1 -c public localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux Ubuntu 2.4.18-3 #1 Thu Apr 18 07:31:07 EDT
2002 i386
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs
SNMPv2-MIB::sysUpTime.0 = Timeticks: (28119) 0:04:41.19
SNMPv2-MIB::sysContact.0 = STRING: Webmaster <Webmaster@enp.edu.dz>
SNMPv2-MIB::sysName.0 = STRING: Ubuntu
SNMPv2-MIB::sysLocation.0 = STRING: CCRSI-ENSP, Alger, Algérie
. . .
```

Affection d'une nouvelle valeur à l'objet sysLocation de la MIB de l'agent avec SNMPv1 :

```
# snmpset -v 1 -c public localhost system.sysLocation.0 s "hello"
Error in packet.

Reason: (noSuchName) There is no such variable name in this MIB.
Failed object: SNMPv2-MIB::sysLocation.0
```

Affection d'une nouvelle valeur à l'objet sysLocation.0 de la MIB de l'agent avec SNMPv2c:

```
# snmpset -v 2c -c public localhost system.sysLocation.0 s "hello"
Error in packet.

Reason: notWritable (that object does not support modification)
Failed object: SNMPv2-MIB::sysLocation.0
```

En comparant les résultats des 2 dernières commandes, on voit que SNMPv1 ne définit pas en retour de code d'erreur, ce que corrige SNMPv2c. Dans notre cas, on essaye de modifier un objet de la MIB de l'agent SNMP distant accessible en lecture seulement !

Conclusion

On a pu voir à travers ce chapitre l'importance de l'administration de réseau qui est une des tâches qui incombe généralement à l'administrateur système pour un réseau local. Pour un réseau étendu, la tâche devient plus ardue avec les contraintes de garder le réseau en état de marche et d'intervenir au plus vite en cas de pannes.

Le protocole SNMP a été développé pour faciliter la supervision des réseaux. On a pu voir que grâce à des requêtes SNMP simples (GET, SET…) et la remontée d'informations par TRAPs SNMP, on pouvait maintenir son réseau en état de bon fonctionnement.

La richesse des informations, pouvant être récupérées à partir des équipements administrés, est directement liée à la complexité de la MIB et la quantité de variables dont elle dispose.

Nous verrons brièvement dans le prochain chapitre comment ces mesures peuvent être organisées et présentées.

CHAPITRE III

Exploitation des résultats de la métrologie passive – Monitoring des réseaux IP

Introduction

On a vu dans le chapitre précédent l'importance et l'efficacité du protocole SNMP dans le cas des mesures passives. Mais ces données récupérées sont instantanées, et pour pouvoir les apprécier correctement, une certaine mise en forme est nécessaire.

Dans ce chapitre, on va introduire la notion de Monitoring ou de supervision par l'exploitation des données récoltées via l'utilisation de SNMP. On introduira les notions fondamentales puis on passera à l'emploi de MRTG, tout en expliquant son fonctionnement. Après cela, on détaillera son installation et sa configuration.

1. Définition de la supervision d'un réseau

Le principe de base de la supervision ou *monitoring* est d'avertir en cas de problème le responsable d'une ressource avant même que les utilisateurs ne s'en aperçoivent. Ce responsable pourra ainsi intervenir dans les plus brefs délais sur un ordinateur ou un service. Ceci permet d'être *proactif* pour un service informatique qui gère de nombreux serveurs sur lesquels résident beaucoup de services. Par exemple, une alerte peut être envoyée dans le cas d'un serveur dont la charge CPU dépasse un certain seuil, qu'un disque en miroir est défectueux ou que l'espace libre sur un disque devient insuffisant. La même chose peut être faite pour surveiller un service comme HTTP, HTTPS, SSH, SMTP, etc. **[III.1]**

2. Monitoring VS surveillance des performances

Un amalgame est souvent fait entre le *monitoring* et le suivi de l'utilisation de ressources. Ceci est probablement dû au fait que certains logiciels commerciaux essaient d'intégrer les deux choses dans le même logiciel. Il est important de bien différencier ces deux aspects, car autant le *monitoring* doit rester simple pour être efficace, autant le suivi de l'utilisation de ressources peut s'avérer beaucoup plus compliqué. Ce suivi est pratique pour le dimensionnement d'un service ou d'un serveur ou pour mettre en évidence un problème de configuration. Par exemple, sur un serveur Web Apache on désire suivre le nombre de requêtes par seconde. Si on les met en relation avec la charge CPU du serveur, on peut se rendre compte à partir de quel moment il devient sous-dimensionné. Cet aspect est aussi important, mais ce n'est pas purement du *monitoring*. **[III.1]**

3. Monitoring avec MRTG-SNMP

3.1 Historique

La première apparition de l'outil MRTG fut avec les essais de Tobias Oetiker qui, pour connaitre le comportement de son réseau, a écrit un script pour mettre à jour avec une fréquence constante un graphe sur le net. Ce graphe montrait le trafic réseau sur un lien

internet. Ce programme a évolué par la suite donnant ainsi lieu à un script Perl configurable appelé MRTG (Multi Route Traffic Grapher)-1.0 en 1995.

Puis, en Janvier 1996, Tobias Oetiker a collaboré avec Dave Rand pour augmenter l'efficacité de MRTG qui était lent du fait qu'il était entièrement écrit en Perl. Cette solution consistait à réécrire les sections critiques traitant la gestion du temps d'exécution de MRTG en Langage C. Comme résultat, la vitesse de MRTG s'est accrue avec un facteur de 40.

Actuellement, MRTG en est à sa version 2 et ne cesse d'être mis à jour compte tenu de sa gratuité et de l'énorme feedback dont il bénéficie. **[III.2]**

3.2 Présentation de MRTG

MRTG **[II.3][II.4][III.2]** est constitué d'un script Perl qui utilise le protocole SNMP pour lire les compteurs de trafic sur un réseau IP, couplé à un programme écrit en langage C. Ce dernier enregistre les données relatives à cette activité des données pour générer des graphes en vue de monitorer cette connexion réseau. Ces graphiques sont intégrés dans des pages HTML qui peuvent être consultées à partir de n'importe quel navigateur Web moderne (IE, FireFox…..).

En plus d'une vue quotidienne, MRTG crée également des représentations visuelles de l'activité du réseau vue au cours des sept derniers jours, cinq dernières semaines et au cours des douze derniers mois. Cela est possible parce que MRTG tient un registre de toutes les données qu'il tire de l'équipement réseau administré. Ce journal est automatiquement consolidé de manière à ne pas croître au fil du temps, mais contient toutes les données pertinentes pour l'ensemble du trafic constaté au cours des deux dernières années.

MRTG n'est pas limité à la surveillance du trafic sur le réseau visé seulement. Il est possible de contrôler les variables SNMP de notre choix. On peut même utiliser un programme externe pour recueillir les données qui doivent être surveillés par MRTG. Les gens utilisent MRTG, pour surveiller des paramètres comme la charge CPU du système, les Sessions d'identification, disponibilité d'un Modem et plus encore. MRTG nous permet même de cumuler deux ou plusieurs sources de données en un seul graphique. **[III.2]**

3.3 Caractéristiques de l'outil MRTG

Portable

MRTG fonctionne sur la plupart des plates-formes UNIX et Windows NT.

Utilisation de Perl

MRTG est écrit en Perl et dispose d'une source complète.

Une identification fiable des interfaces réseau

Les interfaces des routeurs par exemple peuvent être identifiées par des adresses IP, leurs descriptions et leurs adresses MAC en plus des numéros traditionnels de ces interfaces (ex : eth0).

Des fichiers log à taille constante

La taille des fichiers log de MRTG n'augmente pas grâce à l'utilisation d'un algorithme unique pour la consolidation des données.

Configuration automatique

MRTG est livré avec un ensemble d'outils de configuration qui rendent cette dernière et son installation très simple.

Performance accrue

Les routines traitant le temps d'exécution sont écrites en C.

Des graphiques GIF libres

Les graphiques sont générés directement au format PNG en utilisant la bibliothèque GD **[III.3]** de Thomas Boutell.

Personnalisation

L'apparence des pages web produites par MRTG est hautement configurable.

Compatibilité avec RRDTool

Le format de sortie des graphes sous MRTG peut être configuré au format de l'outil RRDTool.

3.4 Principe de stockage des données dans les fichiers log

Un élément clé de MRTG-2 est sa méthode pour le maintien des fichiers log. L'hypothèse de base, lors de la conception du fichier log de MRTG-2, est que l'intérêt pour les informations détaillées sur la charge du réseau, diminue proportionnellement au temps qui s'est écoulé entre la collecte de l'information et son analyse.

Cela a conduit à la création d'un fichier log (journal), qui emmagasine les données, avec une résolution qui décroit en remontant au passé. Ainsi, les données qui datent de plus de deux ans seront supprimées de ce fichier. La résolution de ce dernier correspond à celle des graphiques sur la page Web, actualisée par la suite.

De ce fait, tracer les différents graphiques est relativement rapide, car aucune mesure de réduction des données n'est nécessaire et, par conséquent, les instructions d'E / S disque sont réduites au minimum.

Figure III-1 : Traitement des fichiers log (en ASCII)

Les fichiers log sont stockés en format ASCII. Chaque ligne commence avec une étiquète, relative à la date et l'heure, suivie par les données relatives au trafic. Le fichier commence avec la plus récente entrée et se termine sur deux ans dans le passé. Pour le traitement, il est lu comme un tout, transformés dans la mémoire puis écrit sur le disque. Cela se produit pour chaque mise à jour comme le montre la figure III-1. Quand à la figure III-2, elle indique comment les valeurs du fichier de log sont consolidées au fil du temps. **[III.4]**

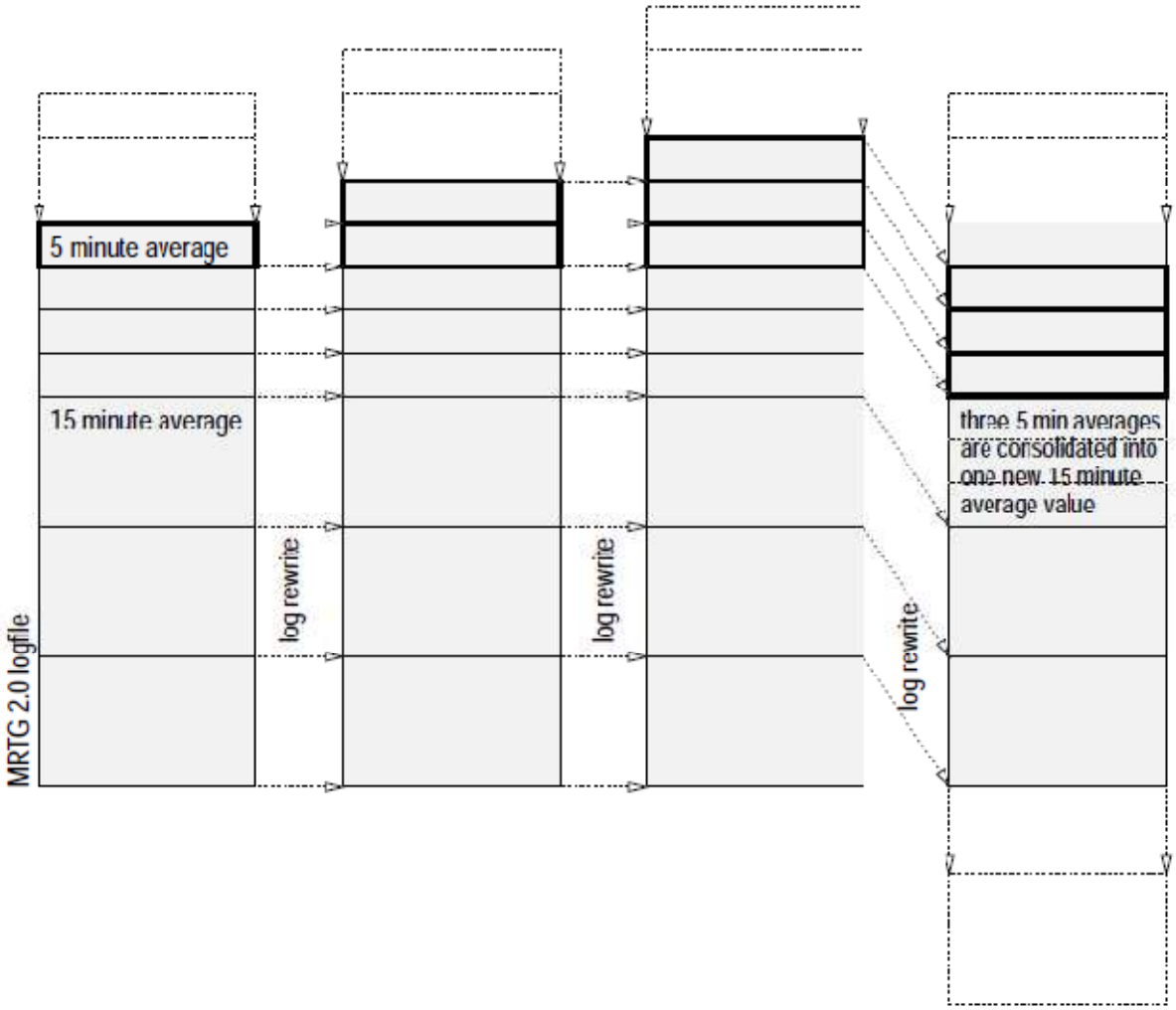


Figure III-2 : Traitement des données dans le fichier log de MRTG-2

4 Mise en œuvre de MRTG

L'un des principaux avantages de MRTG est qu'il est multi plate-forme vu qu'il a été développé en langage Perl qui existe aussi bien sur Linux que sur Windows.

4.1 Sous Windows

4.1.1 Pré-requis

- Installation de l'agent SNMP sur la machine Windows qu'on veut administrer ;
- Installation d'ActivePerl nécessaire au fonctionnement de MRTG ;

4.1.2 Installation

L'installation de SNMP sous Windows étant traitée dans le chapitre précédent, on passera directement à l'intégration d'ActivePerl puis à l'installation de MRTG.

4.1.2.1 Installation d'ActivePerl

Ici, l'installation se fera à partir d'un fichier *EXE* ou un *MSI*. On exécutant le programme on devra aboutir à l'écran suivant qui nous permettra de choisir les composants à installer :

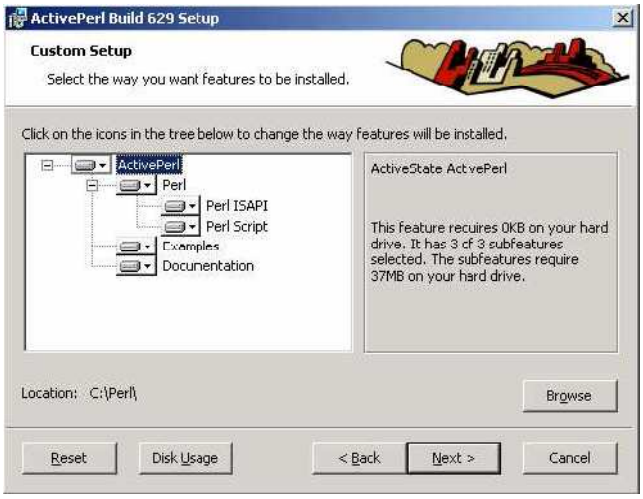


Figure III-3 : installation d'ActivePerl

Vous pourra éventuellement laisser les options par défaut qui installeront la totalité du moteur **Perl** sur l'ordinateur.

A noter que le chemin d'installation présent en bas de la fenêtre est par défaut **C:\Perl**.

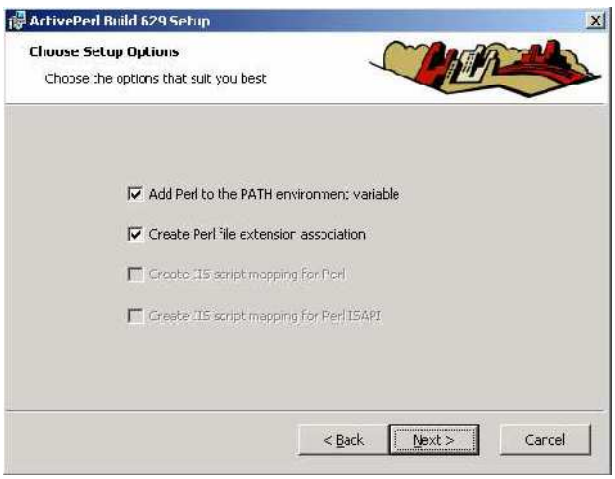


Figure III-4 : ajout de Perl aux variables d'environnement PATH

Remarque

En fonction des composants présents sur la machine locale, il se peut que les options qui sont ici grisées soient actives. Elles permettent de paramétrer IIS afin qu'il puisse être capable d'exécuter des scripts écrits en langage Perl.

Il faut cocher les cases qui nous sont nécessaires sachant que seule la première nous est indispensable.

L'installation terminée, il faudra juste vérifier que le programme d'installation a bien ajouté le chemin **C:\Perl\Bin** dans la variable système **PATH** comme nous l'avions demandé sur l'un des écrans lors de l'installation. Pour cela, faites un clic droit puis **Propriétés** sur le **Poste de Travail**. Dans la boîte de dialogue qui apparaît allez sur l'onglet **Avancé** afin d'obtenir la boîte de dialogue suivante :

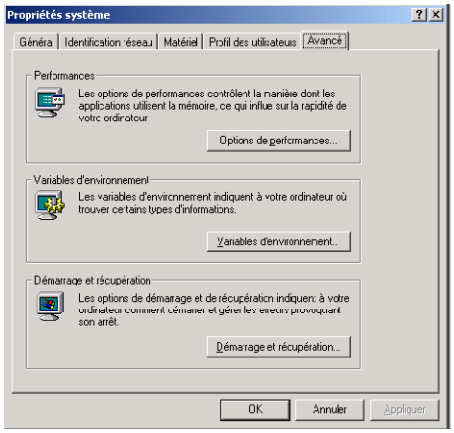


Figure III-5 : Vérification de l'installation

Cliquer sur le bouton **Variables d'environnement** afin d'obtenir ceci :

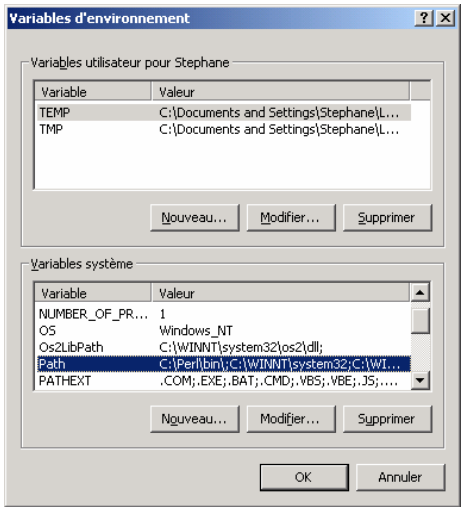


Figure III-6 : Confirmation de l'intégration d'ActivePerl

Si l'installation s'est bien passée comme c'est le cas ici, on voit sur la ligne en surbrillance qu'elle commence par **C:\Perl\Bin**. Si ce n'est pas le cas, il faudra cliquer sur le bouton **Modifier** et ajouter **C:\Perl\Bin;** au tout début de la ligne (sans oublier le point virgule).

4.1.2.2 Installation de MRTG

L'installation de **MRTG** est très simple et très rapide. Le programme d'installation se présente normalement sous la forme d'un fichier **ZIP**. Il faut simplement décompresser ce fichier dans un répertoire par exemple **C:\MRTG**.

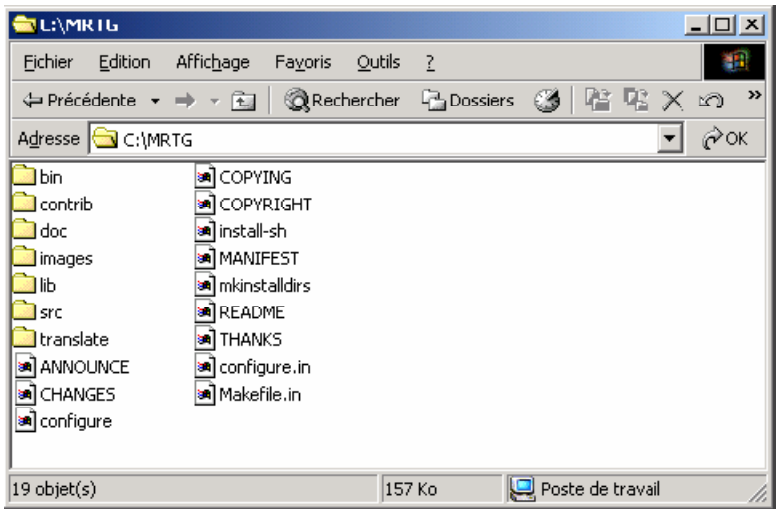


Figure III-7 : répertoire d'installation de MRTG

Note

Comme l'application réalisée pour ce mémoire est faite sur des machines Linux, on parla de configuration MRTG sous Ubuntu uniquement. (Voir Annexe B pour un exemple de configuration sous Windows).

4.2 Sous Linux

4.2.1 Pré-requis

MRTG se contente de construire des pages HTML. Il nous faudra donc disposer de plusieurs choses :

- Un agent SNMP bien sûr, sur l'hôte dont on veut auditer les paramètres ;
- Un serveur HTTP (Apache, par exemple) sur la même machine où on installera MRTG ;
- Quelques paquetages de perl. Mais si, comme nous allons le faire ici, MRTG est installé avec la commande « apt » (qui exige une connexion à Internet), les dépendances seront gérées automatiquement. **[II.9]**

4.2.2 Installation

Sur une machine Ubuntu, la façon la plus simple d'installer MRTG est d'avoir recourt à « apt-get » par la ligne de commandes suivante :

```
# apt-get install mrtg
```

Cette dernière installera le paquetage MRTG par internet sur la machine locale et comme résultats on aura :

Dans le répertoire « /var/www/ »

C'est le répertoire par défaut du serveur APACHE dans lequel sera ajouté un répertoire nommé mrtg qui est le répertoire de travail par défaut de MRTG sous une machine Ubuntu.

Dans le répertoire «/usr/bin/»

C'est le répertoire des exécutables dans lequel s'ajouterons :

- mrtg lui-même, nous verrons quoi en faire,
- cfgmaker, un script qui nous permettra de créer les fichiers de configuration pour MRTG,
- indexmaker, pour générer la génération des pages Web.

4.2.3 Configuration

Pour construire une configuration pour MRTG, il faut :

- Créer un fichier de configuration « .cfg » (ex : mrtg.cfg). Pour cela, il faut utiliser la commande *cfgmaker* dont la forme la plus simple d'utilisation est :

  ```
  # cfgmaker commuanuté@adresse_IP > /chemin/fichier_de_sortie.cfg
  ```

  Le rôle de ce fichier est la description de l'interface réseau désirée (désignée dans ce cas par son adresse IP).

- Générer le fichier *index.html* (sous */var/www/mrtg* car *cfgmaker* est utilisée ici sans options) correspondant à cette configuration avec la commande *indexmaker* :

  ```
  # indexmaker fichier_de_configuration.cfg > /var/www/mrtg/index.html
  ```

  Qui va générer dans ce cas-ci une page Web sur le lien '*http://adresseIP/mrtg*' contenant donc le trafic Ethernet sur cette interface.

Exemple de fichiers de configuration

- Pour un fichier de configuration standard (visualisation du trafic Ethernet) généré par la commande *cfgmaker* sur une machine Linux, les paramètres les plus importants sont les suivants :

```
# Global Config Options

#  for Debian
WorkDir: /var/www/mrtg

### Global Defaults

######################################################################
# System: ubuntu
# Description: Linux ubuntu 2.6.24-19-server #1 SMP Wed Jun 18 15:18:00 UTC 2008 i686
# Contact: Root <root@localhost> (configure /etc/snmp/snmpd.local.conf)
# Location: CCRSI (configure /etc/snmp/snmpd.local.conf)
######################################################################

### Interface 2 >> Descr: 'eth2' | Name: 'eth2' | Ip: '172.16.13.23' | Eth: '00-00-00-00-00-02'
###

Target[localhost_eth2]: #eth2:public@localhost:
SetEnv[localhost_eth2]: MRTG_INT_IP="172.16.13.23" MRTG_INT_DESCR="eth2"
MaxBytes[localhost_eth2]: 1250000
Title[localhost_eth2]: Traffic Analysis for eth2 -- ubuntu
PageTop[localhost_eth2]: <h1>Traffic Analysis for eth2 -- ubuntu</h1>
```

La cible (Target) est très importe pour la configuration. Ici, elle fait référence à l'interface eth2 suivit de la communauté et l'adresse de la machine supervisée.
Quand aux autres paramètres, on remarque qu'ils sont en HTML et définissent la forme des pages Web générées.

- Et comme MRTG se base sur des requêtes SNMP, son utilisation s'étend à d'autres caractéristiques propres comme la charge CPU, l'utilisation de la RAM, le pourcentage d'espace disque occupé, etc.
  Pour ce faire, il suffit juste de connaitre le bon paramètre à consulter via *snmpget* sur la MIB et de spécifier son OID correspondant comme cible (Target) dans le fichier de configuration. (Voir Annexe C pour des exemples de configuration)

Conclusion

MRTG a été, dans un premier temps, conçu comme application à usage personnel. Mais sa publication sur Internet, a montré qu'il y avait une demande considérable pour un tel programme.

Beaucoup d'outils libres comme NET-SNMP étaient disponibles pour la récupération des données sur l'état actuel d'un lien réseau. Mais l'utilisation simple de MRTG, avec son approche innovante d'analyse à long terme, et sa présentation conviviale des mesures à travers des pages Web, ont conduit la notion de métrologie des réseaux à un niveau supérieur, d'où le Monitoring.

Quelque défauts existent encore sur la version 2 de MRTG, comme l'incapacité d'afficher des valeurs négatives et le nombre limité d'équipements (pas plus de 600) qu'on peut gérer à la fois. Ces défauts sont notamment corrigés avec la version 3 qui intègre directement le module RRDtools.

Malgré cela, la solution MRTG-SNMP reste idéale pour des réseaux de petites entreprises ou de campus universitaires. Nous nous baserons sur cette dernière pour mettre en place notre application.

CHAPITRE IV

Application : Mise en place de la plate-forme de Métrologie passive

Introduction

Dans les chapitres précédents, on a vu ce qu'il y a d'essentiel pour la supervision d'un réseau IP, à savoir, la description de ses métriques puis le choix des outils adéquats afin de pouvoir les apprécier.

Le choix des mesures passives nous a conduit à utiliser le standard MRTG-SNMP afin de récupérer de façon périodique les valeurs significatives liées au comportement du réseau, puis de les cumuler de façon logique pour enfin les afficher sous forme de graphes stockés dans des pages Web.

De ce fait, on détaillera tout au long de ce dernier chapitre notre projet. Celui-ci consiste en la mise en œuvre d'une plate-forme de métrologie passive afin d'administrer et veiller au bon fonctionnement du réseau d'un campus universitaire donné.

1. Description du réseau

Le réseau sur lequel à été réalisée l'application possède une connexion à Internet à travers un routeur CISCO 2800.

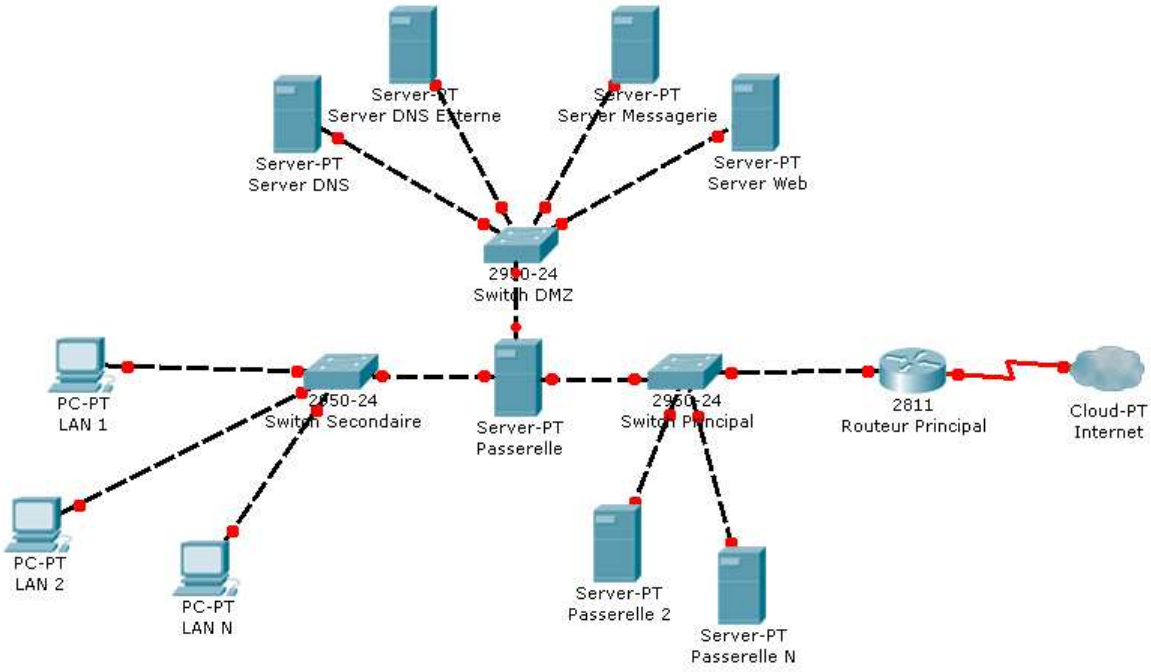


Figure IV-1 : Architecture du réseau supervisé

Il est constitué des éléments suivants :

- Un switch principal de type Catalyst 2950 qui relie les différentes passerelles existantes au routeur principal.
- Un Switch pour les DMZs : interne et externe. Ces zones hébergent les serveurs de ce réseau à savoir : DNS interne, Messagerie, Web, DNS externe.
- Un switch secondaire comportant plusieurs « VLAN » pour la séparation virtuelle entre les différents réseaux locaux (LAN) qui peuvent par exemple représenter les départements du campus en question.
- Une ou plusieurs passerelles dont le rôle est de gérer la sécurité, notamment les différentes règles d'accès et de communications entre les hôtes et les services de ce réseau. Comme gestion de sécurité, on peut parler à titre d'exemple d'un filtrage multi-niveaux (ex : par adresse MAC ou par port TCP/UDP) à l'aide d'un pare-feu.

L'ensemble des switchs cités ci-dessus sont reliés à la passerelle (Figure IV-1) ou bien à plusieurs passerelles suivant la dimension et les nécessités des utilisateurs de ce réseau.

2. Travail à effectuer

Afin de diagnostiquer ce réseau d'une manière correcte, on s'est proposé de mettre en place un serveur dédié au suivi des paramètres suivants :

- Le trafic Ethernet sur les switchs : principal et secondaire afin de mesurer le débit avec lequel transitent les données respectivement au niveau des passerelles et de chaque département.
- Le trafic sortant/entrant au niveau du routeur principal et donc le débit Internet réellement mis à profit par le campus universitaire.
- Le trafic de données par port TCP/UDP (trafic protocolaire) et cela pour en extraire le type et les caractéristiques du trafic IP global auquel on a affaire. Cela nous servira aussi à déceler les anomalies liées aux différents services disponibles comme par exemple une activité Mail anormalement élevée dont la cause peut être un trop grand nombre de SPAM's au niveau du serveur de messagerie.
- Le pourcentage de mémoire physique utilisée sur la passerelle afin d'en prévoir la saturation et éviter ainsi un disfonctionnement.

Mais pour y arriver, tous ces équipements doivent être administrables. Il est donc nécessaire que chacun de ces derniers supporte le protocole SNMP.

3. Réalisation de la plate-forme

Cette partie passe par deux étapes importantes :

a) La configuration des équipements cités précédemment par l'activation ou l'installation de l'Agent SNMP sur chacun de ces derniers ;
b) L'installation de MRTG-SNMP sur le serveur de monitoring dans le but de créer l'interface de supervision qui sera donc sous forme d'une page HTML ;

3.1 Configuration des équipements

3.1.1 Routeur principal

L'accès à cet équipement se fait par câble console afin d'ouvrir une session Telnet et ainsi communiquer avec ce dernier (Figure IV-2).

Figure IV-2 : Accès au routeur par câble console

Il faudrait par la suite activer l'agent SNMP déjà installé sur le routeur à travers les commandes appropriées de l'IOS comme suit :

```
CISCO > enable
CISCO # configure terminal
CISCO (config)#snmp-server community secret ro
CISCO (config)# exit
CISCO# wr
CISCO# copy running-config startup-config
```

- La commande « enable » ou « en » est utilisée pour accéder en mode privilégié d'où l'apparition du « # » à la place du prompt.
- La commande « configure terminal » ou « conf t » permet d'accéder en mode configuration du routeur.
- L'instruction « snmp-server » est quand à elle la plus importante dans ce contexte. Elle nous permet d'activer l'Agent SNMP de cet équipement tout en spécifiant la communauté (*secret* dans cet exemple) mais aussi les droits d'accès « *ro* » (readonly ou lecture seule dans ce cas) à la MIB CISCO locale.

Quand aux commandes « wr » et « copy running-config startup-config », elle permettent respectivement de sauvegarder la configuration actuelle puis de la mettre au niveau des paramètres chargés au démarrage de l'équipement.

3.1.2 Switch principal et secondaire

Dans le cas d'un commutateur Ethernet, la manipulation demande un peu plus d'étapes à accomplir. Cela vient du fait que la notion d'adresse IP n'est pas présente par défaut sur un équipement actif de niveau OSI II alors que SNMP se base sur UDP/IP (voir chapitre II, §2)

Pour remédier à ce problème, il faudrait exploiter la notion de VLAN en utilisant un de ceux qui sont déjà présents au niveau de ce Switch. En cas d'absence de VLAN, on devra en créer un qui sera dédié au monitoring des ports Ethernet.

Là aussi, la configuration à été faite via câble console et les commandes IOS doivent être exécutées dans l'ordre suivant :

```
CISCO > en
CISCO# conf t
CISCO (config)# interface vlan1
CISCO (config-if)# ip address adresse_ip_switch_principal masque_de_sous-réseau
CISCO (config-if)# exit
CISCO (config)#snmp-server community secret2 ro
CISCO (config)# exit
CISCO# wr
CISCO# copy running-config startup-config
```

Donc ce qu'on ajoute dans ce cas, par rapport à la configuration du routeur, sont les commandes :

- « interface » ou « int » avec laquelle on accède au paramétrage du VLAN (vlan1 dans ce cas précis) voulu ;
- « ip address » qui permettra d'attribuer une adresse IP suivie d'un masque de sous-réseau au VLAN de monitoring ;

Remarque :

La procédure de configuration étant ici identique, que ce soit le cas du commutateur principal ou secondaire, l'intérêt de la supervision des deux Switchs réside dans le fait d'apprécier le trafic Ethernet liant les passerelles mais aussi le débit propre à chaque département.

Cela permet au final d'avoir un aperçu global du comportement de notre réseau.

3.1.3 configuration de la passerelle

Pour le cas de la passerelle, les paramètres pour lesquels on a choisi de faire le suivi sont : l'utilisation de la mémoire physique et le trafic par port (TCP/UDP).

La première variable existe dans la MIB de cette passerelle alors que la notion de paquets de données par service (TCP/UDP) n'y figure pas.

La solution proposée consiste à utiliser un script écrit en langage perl combiné avec un firewall (Netfilter de Linux et l'outil shorewall).

Ceci doit se faire localement d'où la nécessité d'installer MRTG en plus de l'Agent SNMP sur cette machine.

Les étapes de réalisation sont :

3.1.3.1 Installation et configuration de l'Agent SNMP

La passerelle sur laquelle on a travaillé est une machine RedHat Enterprise 4. L'installation de NET-SNMP sur cette dernière à été réalisée par l'utilisation des paquetages RPM.

De ce fait, le cheminement est le suivant :

1) On vérifie d'abord qu'aucune installation SNMP n'est présente sur le système avec la commande :

```
# rpm -qa | grep snmp
```

Cette requête ne doit pas retourner de résultat pour confirmer l'absence de SNMP.

2) On installe après le paquetage « net-snmp.rpm » et ses dépendances qui sont des plugins et librairies nécessaires à son intégration sur la machine :

```
# rpm -ivh net-snmp-libs-5.1.2-13.el4.i386.rpm
# rpm -ivh net-snmp-perl-5.1.2-13.el4.i386.rpm
# rpm -ivh net-snmp-utils-5.1.2-13.el4.i386.rpm
# rpm -ivh beecrypt-devel-3.1.0.6.i386.rpm
# rpm -ivh lm-sensors-2.8.7-2.40.5.i386.rpm
# rpm -ivh net-snmp-5.1.2-13.el4.i386.rpm
```

On peut trouver ces fichiers « rpm » dans le DVD d'installation de CentOS v4.7 ;

3) Une fois l'installation menée à terme, on passe à la configuration de l'Agent « NET-SNMP », désigné par « snmpd », en éditant en premier lieu le fichier « snmpd.conf » par la commande :

```
# vi /etc/snmp/snmpd.conf
```

Les changements suivants sont nécessaires :

a) La ligne : `com2sec notConfigUser default public`

Doit être remplacée par :

```
com2sec local localhost secret3
com2sec myManager adresse_IP_serveur_monitoring secret3
```

b) Les lignes :

```
group       notConfigGroup       v1       notConfigUser

group       notConfigGroup       v2c      notConfigUser
```

Doivent devenir :

```
group       MyRWGroup       v1          local

group       MyRWGroup       v2c         local

group       MyRWGroup       usm         local

group       MyROGroup       v1        myManager

group       MyROGroup       v2c       myManager

group       MyROGroup       usm       myManager
```

c) On cherche après cela la ligne :

```
view            systemview            included            system
```

qu'on substitue par :

```
view                all                 included              .1                  80
```

Pour nous autoriser à parcourir la MIB locale depuis le plus haut niveau hiérarchique disponible (*.iso).*

d) Maintenant on attribue les droits de lecture/écriture pour la machine « local » et de lecture seule pour la machine de monitoring (Manager). Pour cela, la ligne :

```
access    notConfigGroup    ""    any    noauth    exact    systemview    none    none
```

Devient:

```
access    MyROGroup    ""    any    noauth    exact        all          none    none
access    MyRWGroup    ""    any    noauth    exact        all           all    none
```

e) On peut éventuellement configurer la localisation de la machine et de contact en éditant les lignes :

```
syslocation     Unknown         (edit /etc/snmp/snmpd.conf)
syscontact      Root            (configure /etc/snmp/snmp.local.conf)
```

Qui deviendront :

```
syslocation     lieu_du_campus_universitaire

syscontact      webmaster       <adresse_mail_du_superviseur>
```

4) Une fois que ces changements seront appliqués, par la sauvegarde du fichier de configuration « snmpd.conf », il sera nécessaire de redémarrer le service « snmpd » via la ligne de commandes suivante :

```
#  /etc/init.d/snmpd restart
```

5) Il faut ensuite s'assurer que ce service démarre automatiquement à chaque réinitialisation de Linux. Pour cela, les commandes suivantes doivent être validées :

```
#  chkconfig snmpd  on
#  service snmpd  start
```

Remarque :

Si la passerelle est une machine Linux (Ubuntu), l'installation de l'Agent SNMP sur cette dernière se fera comme indiqué dans le chapitre II (voir §11.2.1).

3.1.3.2 Préparation de la passerelle pour le monitoring de l'espace disque utilisé

Avant de pouvoir superviser l'utilisation de l'espace disque, il faudrait d'abord voir l'ensemble des partitions présentes en exécutant la commande :

```
#  df
```

Il faut par la suite éditer une nouvelle fois le fichier de configuration « snmpd.conf » en ajoutant la ligne :

```
disk        /       100000000000
```

Cette dernière va permettre à l'Agent SNMP de bénéficier des informations propres à la partition « root » ou « / »

3.1.3.3 Préparation de la passerelle pour le monitoring par port

Le rôle de la passerelle étant le filtrage, un firewall (Netfilter géré par shorewall) existe déjà dessus.

La solution appliquée dans ce cas consiste à utiliser ce pare-feu pour placer des compteurs de trafic afin de surveiller le nombre de paquets de données circulants à travers les interfaces de cette passerelle.

Puis on utilisera un script, écrit en Perl, qui va permettre à MRTG de récupérer les valeurs liées à ce trafic réseau afin de les stocker dans ses propres fichiers « .log ».

Le cheminement des étapes est le suivant :

On crée d'abord un fichier « accounting », ayant pour rôle le comptage du trafic par port, qu'on placera sous ' */etc/shorewall/*'. On supposera ici que la passerelle possède au moins deux interfaces Ethernet :

- eth0 : qui est la carte réseau reliée au switch principal (à Internet donc) ;
- eth1 : qui lie la passerelle au switch secondaire (au réseau local donc) ;

Les lignes qui doivent apparaître dans ce fichier sont les suivantes :

```
#
# /etc/shorewall/accounting
#
#ACTION      CHAIN    SOURCE    DESTINATION    PROTOCOL    DEST    SOURCE
                                                           PORT    PORT

 web:COUNT    -        -         eth0           tcp         -       80
 web:COUNT    -        eth0      -              tcp         80
 web:COUNT    -        eth0      -              tcp         443
 web:COUNT    -        -         eth0           tcp         -       443
 DONE         web

 smtp:COUNT   -        eth0      -              tcp         25
 smtp:COUNT   -        -         eth0           tcp         -       25
 DONE         smtp

 pop3:COUNT   -        eth0      -              tcp         110
 pop3:COUNT   -        -         eth0           tcp         -       110
 DONE         pop3

 imap:COUNT   -        eth0      -              tcp         143
 imap:COUNT   -        -         eth0           tcp         -       143
 DONE         imap

 ssh:COUNT    -        eth0      -              tcp         22
 ssh:COUNT    -        -         eth0           tcp         -       22
 DONE         ssh
```

Ce fichier va permettre de faire le suivie des services Web (HTTP et HTTPS), SMTP, POP3, IMAP et SSH en récoltant le nombre et la taille des paquets de données propres à chacun de ces services.

Une fois ce fichier sauvegardé, il faudra ensuite redémarrer le firewall par la commande :

```
# /etc/init.d/shorewall restart
```

On peut éventuellement vérifier le bon fonctionnement de ces compteurs en exécutant la commande :

```
# shorewall show accounting nom_du_service
```

Qui affichera donc le nombre et la taille globale des paquets de données par port TCP (ou UDP).

On passera en dernier lieu à la création du script « shorewall_stats.pl » (qu'on placera dans un endroit convenable (ex : */usr/bin/*)) qui contient les lignes de configuration suivantes :

```
#!/usr/bin/perl
#####################################################################
#
#   Shorewall Accounting Statistics for MRTG
#   http://www.tylerdavis.com/code/
#
#   This script is under the GPL (GENERAL PUBLIC LICENSE)
#   http://www.gnu.org/copyleft/gpl.html
#
#   (C) 2004 by Tyler Davis <me@tylerdavis.com>
#
#####################################################################
# CONFIGURATION
#####################################################################
# path to shorewall binary
$shorewall = "/sbin/shorewall";
# path to uptime binary
$uptime    = "/usr/bin/uptime";
# path to hostname or simply enter hostname
$hostname  = "/bin/hostname";
#####################################################################
# END OF CONFIGURATION
#####################################################################

      # see if any counters were defined
      if (!@ARGV) {
            print "\nShorewall Accounting Statistics for MRTG\n";
            print "Version v0.0.1\n";
            print "http://www.tylerdavis.com/code/\n\n";
```

```
            print "This script is under the GPL (GENERAL PUBLIC
LICENSE)\n";
            print "http://www.gnu.org/copyleft/gpl.html\n\n";
            print "(C) 2004 by Tyler Davis <me\@tylerdavis.com>\n\n";
            print "Usage: shorewall_stats.pl <counter>\n\n";
            exit;
         }

# get counter and hostname
$show   = $ARGV[0];
$host   = `$hostname`;

# get in/out statistics
$in   = `$shorewall show $show | head -n 7 | tail -1 | awk '{ print \$2
}'`;
$out  = `$shorewall show $show | head -n 8 | tail -1 | awk '{ print \$2
}'`;

#Convert data transferred to bytes
if ( rindex($in,"M") != "-1") {
      #Megabytes
      $in = ($in*1024*1024);
   } else {
      #Kilobytes
      $in = ($in*1024);
   }
if ( rindex($out,"M") != "-1") {
        #Megabytes
        $out = ($out*1024*1024);
   } else {
        #Kilobytes
        $out = ($out*1024);
   }

$uptime = `$uptime`;
$uptime =~ /up (.*?,.*?),/;
$up = $1;

# print out data in an mrtg friendly way
print "$in\n";
print "$out\n";
print "$up\n";
print "$host";
# END
```

Et on finie par l'attribution des droits d'exécution à ce ficher par l'instruction :

```
#   chmod a+x  /usr/bin/shorewall_stats.pl
```

3.2 Mise en place du service de monitoring

La plate-forme mise en place pour la supervision de ce réseau est une combinaison entre une machines Linux, qu'on définie comme serveur de monitoring, et la passerelle qui inclue elle-même le service MRTG.

De ce fait, on montrera les différents paramétrages appliqués à ces deux entités :

3.2.1 Paramétrage du serveur de monitoring (Manager)

Ce serveur est une machine Ubuntu sur laquelle on a installé NET-SNMP (voir chapitre II, §11.2) et MRTG (voir chapitre III, §4.2).

Son rôle est de :

- afficher les trafics Ethernet des équipements CISCO, sur lesquels on a installé un Agent SNMP, à savoir le routeur principal et les deux switchs : principal et secondaire ;
- afficher l'occupation de l'espace disque au niveau de la passerelle ;

L'idée est de générer une page principale (Accueil) qui contient les dossiers « routeur-principal », « sw-pr », « sw-sec » et « passerelle » propres à chacun des éléments réseau qu'on a administré. On accède ensuite aux différents graphes à travers ces dossiers.

Le paramétrage a été fait en suivant les étapes ci-dessous :

1) Création des répertoires de travail sous « */var/www/mrtg/* » par la ligne des commandes :

```
# mkdir /var/www/mrtg/routeur-principal && mkdir /var/www/mrtg/sw-pr
# mkdir /var/www/mrtg/sw-sec && mkdir /var/www/mrtg/passerelle
```

2) Génération des fichiers « cfg », propres aux équipements CISCO supervisés, dans leurs répertoires respectifs :

```
# cfgmaker --global 'WorkDir: /var/www/mrtg/routeur-principal'
  secret@adresse_IP_routeur-principal > /var/www/mrtg/routeur-principal/routeur-
  principal.cfg
# cfgmaker --global 'WorkDir: /var/www/mrtg/sw-pr'
```

```
   secret@adresse_IP_switch-principal > /var/www/mrtg/sw-pr/sw-pr.cfg
# cfgmaker --global 'WorkDir: /var/www/mrtg/sw-sec'
   secret@adresse_IP_switch-secondaire > /var/www/mrtg/sw-sec/sw-sec.cfg
```

3) Génération du fichier « passerelle.cfg », permettant de voir l'occupation de l'espace disque sur celle-ci, avec la commande vi :

```
# vi /var/www/mrtg/passerelle/passerelle.cfg
```

Les lignes de configuration à ajouter sont :

```
WorkDir: /var/www/mrtg/passerelle
LoadMIBs: /usr/share/snmp/mibs/UCD-SNMP-MIB.txt
  Target[DiskRootPercent]:dskPercent.1&dskPercent.1:secret3@adresse_IP_passerelle
  RouterUptime[DiskRootPercent]: secret3@adresse_IP_passerelle
  MaxBytes[DiskRootPercent]: 100
  Title[DiskRootPercent]: DISK USAGE
  PageTop[DiskRootPercent]: <H1>DISK / Usage %</H1>
  Unscaled[DiskRootPercent]: ymwd
  ShortLegend[DiskRootPercent]: %
  YLegend[DiskRootPercent]: DISK Utilization
  Legend1[DiskRootPercent]: Root disk
  LegendI[DiskRootPercent]:  Root disk
 Options[DiskRootPercent]: growright,gauge,nopercent
```

4) Génération des pages HTML pour chaque service proposé :

```
# cd /var/www/mrtg/routeur-principal
# indexmaker routeur-principal.cfg > index.html
# LANG=C mrtg routeur-principal.cfg
# cd /var/www/mrtg/sw-pr
```

```
# indexmaker sw-pr.cfg > index.html
# LANG=C mrtg sw-pr.cfg
# cd /var/www/mrtg/sw-sec
# indexmaker sw-sec.cfg > index.html
# LANG=C mrtg sw-sec.cfg
# cd /var/www/mrtg/passerelle
# indexmaker passerelle.cfg > index.html
# LANG=C mrtg passerelle.cfg
```

5) Planification du rafraichissement, par 5mn d'intervalle, de ces pages Web en éditant le fichier « crontab » :

```
# vi /etc/crontab
```

Les lignes suivantes doivent être ajoutées dedans :

```
0-59/5 * * * * root /usr/bin/mrtg /var/www/mrtg/routeur-principal/routeur-principal.cfg
0-59/5 * * * * root /usr/bin/mrtg /var/www/mrtg/sw-pr/sw-pr.cfg
0-59/5 * * * * root /usr/bin/mrtg /var/www/mrtg/sw-sec/sw-sec.cfg
0-59/5 * * * * root /usr/bin/mrtg /var/www/mrtg/passerelle/passerelle.cfg
```

3.2.2 Paramétrage du service de monitoring par port sur la passerelle

Ici, le fichier que doit cibler MRTG est un script Perl qui doit être utilisé en local. Pour cette raison, on a choisi d'installer ce service aussi sur la passerelle.

Cette installation nécessite les étapes suivantes :

1) Installation du serveur Web « APACHE » par le biais des paquetages « rpm » :

```
# rpm -ivh apr-0.9.4-24.9.i386.rpm
# rpm -ivh apr-util-0.9.4-22.el4.i386.rpm
# rpm -ivh httpd-suexec-2.0.52-9.ent.i386.rpm httpd-2.0.52-9.ent.i386.rpm
```

2) Installation de MRTG, aussi avec les paquetages rpm :

```
# rpm -ivh gd-2.0.28-5.4E.el4_6.1.i386.rpm
# rpm -ivh mrtg-2.10.15-2a.i386.rpm
```

3) Création d'un répertoire de travail pour le stockage des graphes :

```
# mkdir -p /var/www/html/passerelle/
```

4) Création du fichier de configuration « .cfg » pour le monitoring par port (par service) :

```
# vi /var/www/html/passerelle/passerelle_port_traffic.cfg
```

Les lignes de configuration à ajouter sont :

```
WorkDir: /var/www/html/passerelle

###-------------- HTTP/HTTPS TRAFFIC --------------###
Target[http]: `perl /usr/bin/shorewall_stats.pl web`
MaxBytes[http]: 12500000
Title[http]: HTTP Traffic
PageTop[http]: <H1>HTTP Traffic</H1>
LegendI[http]:  HTTP Out:
LegendO[http]:  HTTPS Out:

###-------------- SMTP TRAFFIC ----------------------###
Target[smtp]: `perl /usr/bin/shorewall_stats.pl smtp`
MaxBytes[smtp]: 12500000
Title[smtp]: SMTP Traffic
PageTop[smtp]: <H1>SMTP Traffic</H1>

###-------------- POP3 TRAFFIC ----------------------###
Target[pop3]: `perl /usr/bin/shorewall_stats.pl pop3`
MaxBytes[pop3]: 12500000
Title[pop3]: POP3 Traffic
PageTop[pop3]: <H1>POP3 Traffic</H1>
```

```
###--------------  IMAP TRAFFIC  -----------------------###
Target[imap]: `perl /usr/bin/shorewall_stats.pl imap`
MaxBytes[imap]: 12500000
Title[imap]: IMAP Traffic
PageTop[imap]: <H1>IMAP Traffic</H1>


###--------------  SSH TRAFFIC  -----------------------###
Target[ssh]: `perl /usr/bin/shorewall_stats.pl ssh`
MaxBytes[ssh]: 12500000
Title[ssh]: SSH Traffic
PageTop[ssh]: <H1>SSH Traffic</H1>
```

5) Génération de la page Web correspondante à cette configuration :

```
#  cd  /var/www/html/passerelle
#  indexmaker  passerelle_port_traffic  >  index.html
#  cp  -av  /var/www/mrtg/*.png  /var/www/html/passerelle
#  LANG=C  mrtg  /var/www/html/passerelle/passerelle_port_traffic.cfg
```

6) Et enfin, la planification du rafraichissement (par 5mn d'intervalle) de cette page Web en éditant le fichier « crontab » :

```
#  vi  /etc/crontab
```

La ligne suivante doit être ajoutée dedans :

```
*/5 * * * *  root  /usr/bin/mrtg  /var/www/html/passerelle/parrerelle_port_traffic.cfg
--logging /var/log/mrtg.log
```

Voilà, à ce stade, toutes les configurations ont été faites. Il ne reste plus qu'à attendre la première génération des pages HTML par MRTG.

4. Exploitation des résultats obtenus

4.1 Sur le serveur de monitoring

L'interface de monitoring que nous avons mis en place se présente comme une page Web (HTML) qu'on peut consulter sur l'adresse *http://Adresse_IP_serveur_monitoring/mrtg* via un navigateur Web évolué (FireFox, IE ...).

Sur celle-ci, on trouvera les quatre répertoires relatifs aux équipements supervisés (cf. figure VI-3)

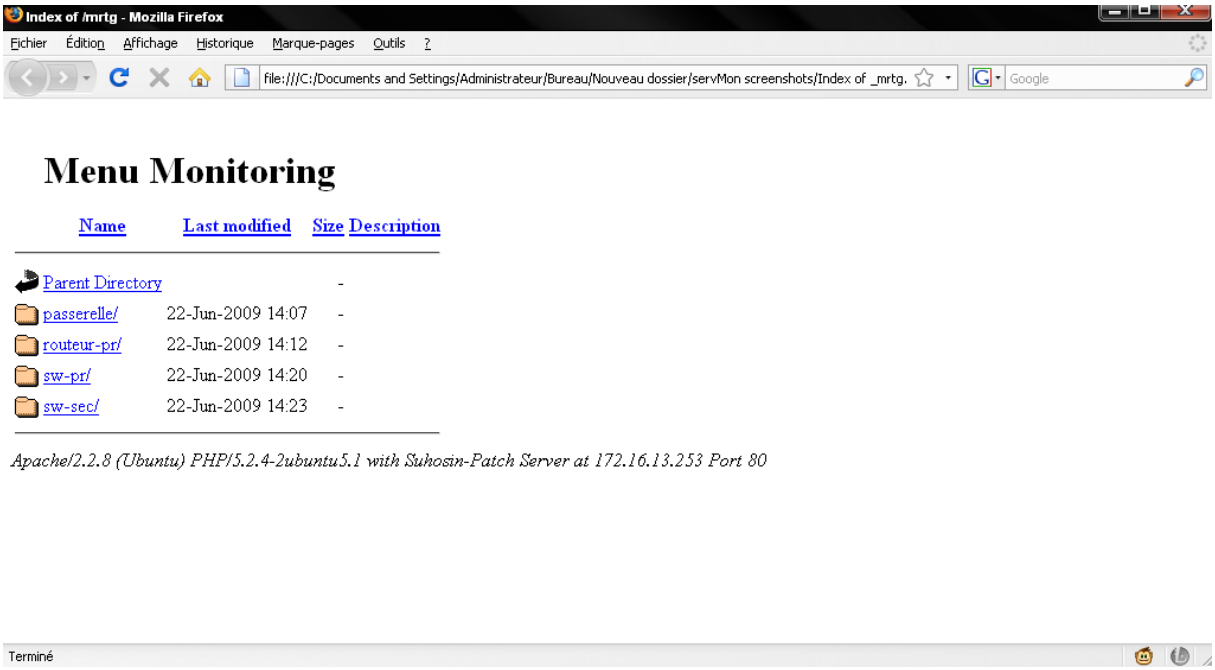


Figure VI-3 : Interface de Monitoring

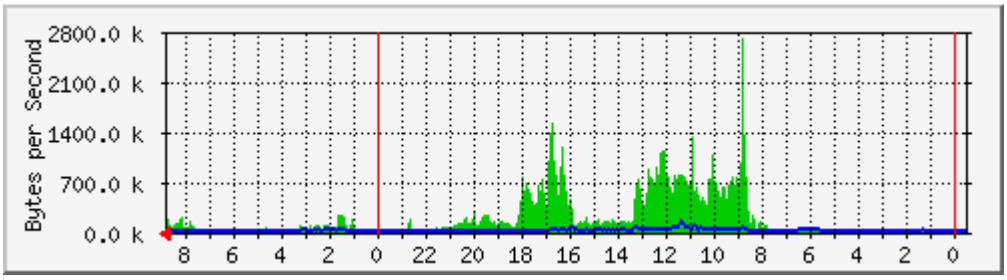
Ces répertoires contiennent à la fois :

- Les fichiers log où les données sont stockées ;
- Le fichier cfg que nous avons créé durant la manipulation ;
- Le fichier « index.html » pour l'affichage des pages Web ;
- Les fichiers images (PNG) contenant les différents graphes tracés par MRTG ;

Pour le cas des équipements CISCO, à savoir le routeur et les switchs, on trouvera dans leurs répertoires respectifs les graphes correspondants au trafic Ethernet à travers chacune de leurs interfaces.

On donnera ici l'exemple du routeur principal où les interfaces visualisées sont : le port Giga Ethernet, relié au réseau du campus universitaire, et le port Serial connecté au WAN. (cf. figure VI-4)

Trafic Ethernet sur Se0/0 du Routeur Principal



Trafic Ethernet sur Gi0/0 du Routeur Principal

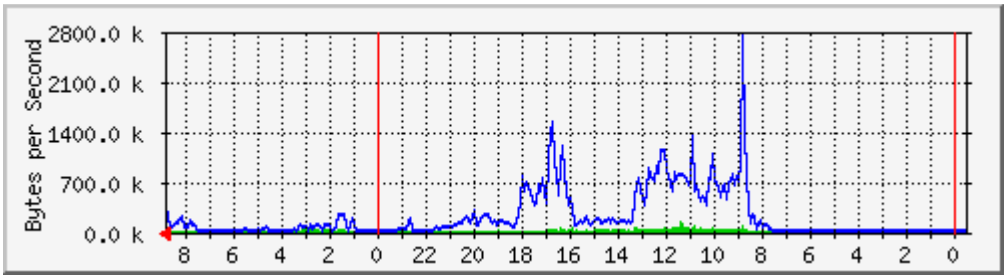
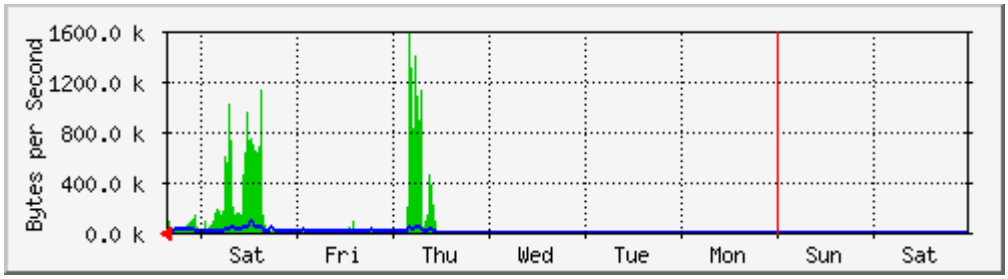


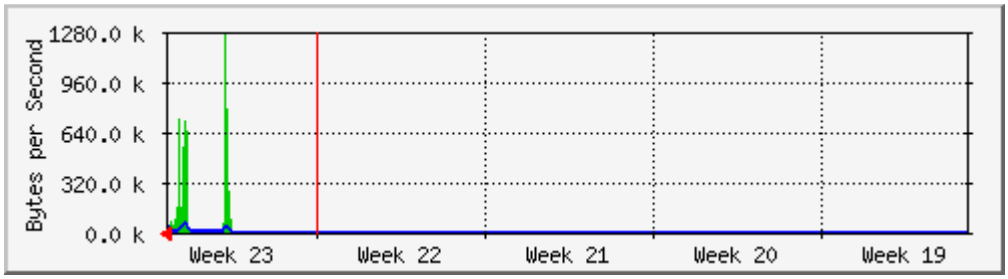
Figure IV-4 : Graphes MRTG journaliers pour le trafic Ethernet sur Gi0/0 et Se0/0 du routeur principal

Ces graphes sont des graphes journaliers (daily) exprimant le débit, entrant (vert) et sortant (bleu) en Octet. En cliquant sur l'un de ces graphes, on accède à une autre page Web qui montre cette fois : le trafic journalier, hebdomadaire, mensuel et annuel. On choisit le cas de l'interface Se0/0 pour voir ce que ça donne (cf. figure IV-5).

`Daily' Graph (5 Minute Average)



`Weekly' Graph (30 Minute Average)



`Monthly' Graph (2 Hour Average)

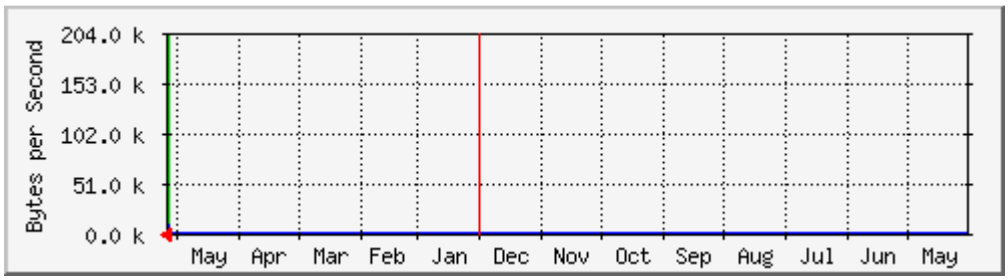


Figure IV-5 : Graphe MRTG journalier, hebdomadaire et mensuel pour le trafic Ethernet sur Se0/0 du routeur principal

Idem pour le switch secondaire, sur lequel on voie le trafic Ethernet journalier au niveau de ses ports Fast Ethernet Actifs (cf. figure IV-6).

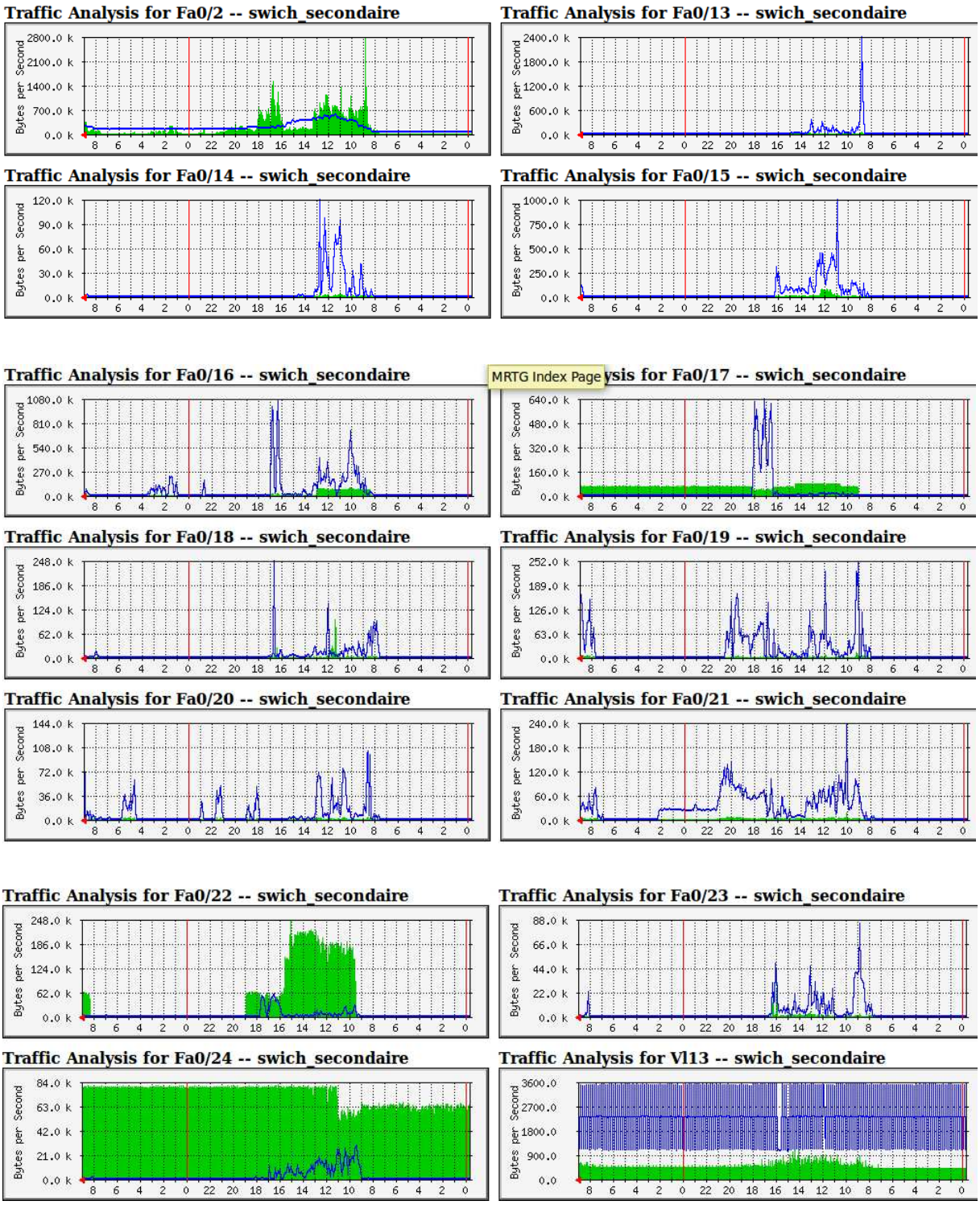


Figure IV-6 : Graphe MRTG journaliers pour le trafic Ethernet sur l'ensemble des ports actifs sur le switch secondaire

On remarquera qu'on visualise aussi le trafic par VLAN, illustré ici par le Vl13.

Par ces données, on peut faire le suivi de la bande passante occupée par chaque département de ce campus universitaire.

Quand au répertoire « passerelle », on trouve dedans les graphes concernant le pourcentage d'espace disque utilisé par cette dernière. (cf. figure IV-7)

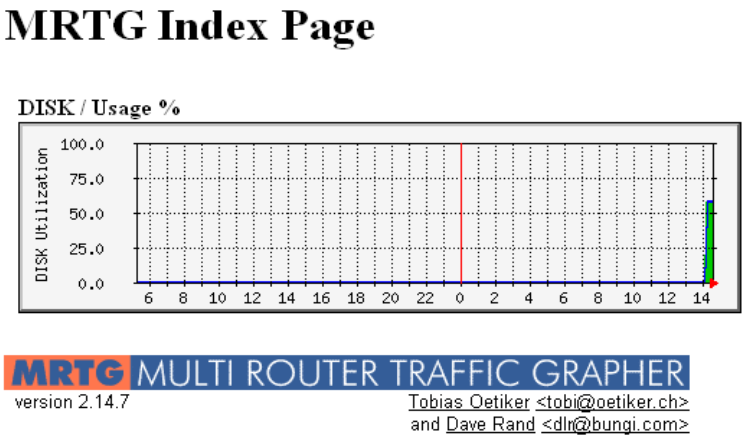


Figure IV-7 : Graphe MRTG pour l'espace disque occupé sur la passerelle

4.2 Sur la passerelle

Afin de consulter le trafic par port sur la passerelle, il faut y accéder via l'adresse *http://Adresse_IP_passerelle/passerelle*.

Les graphes présents sur cette page HTML présentent les trafics de données propres à chaque service qu'on a spécifié dans la configuration du fichier *accounting* (§3.2.2).

Parmi les plus importants, on montrera ici les trafics relatifs au service Web (HTTP/HTTPS) et SMTP (Mail). (cf. figure VI-8)

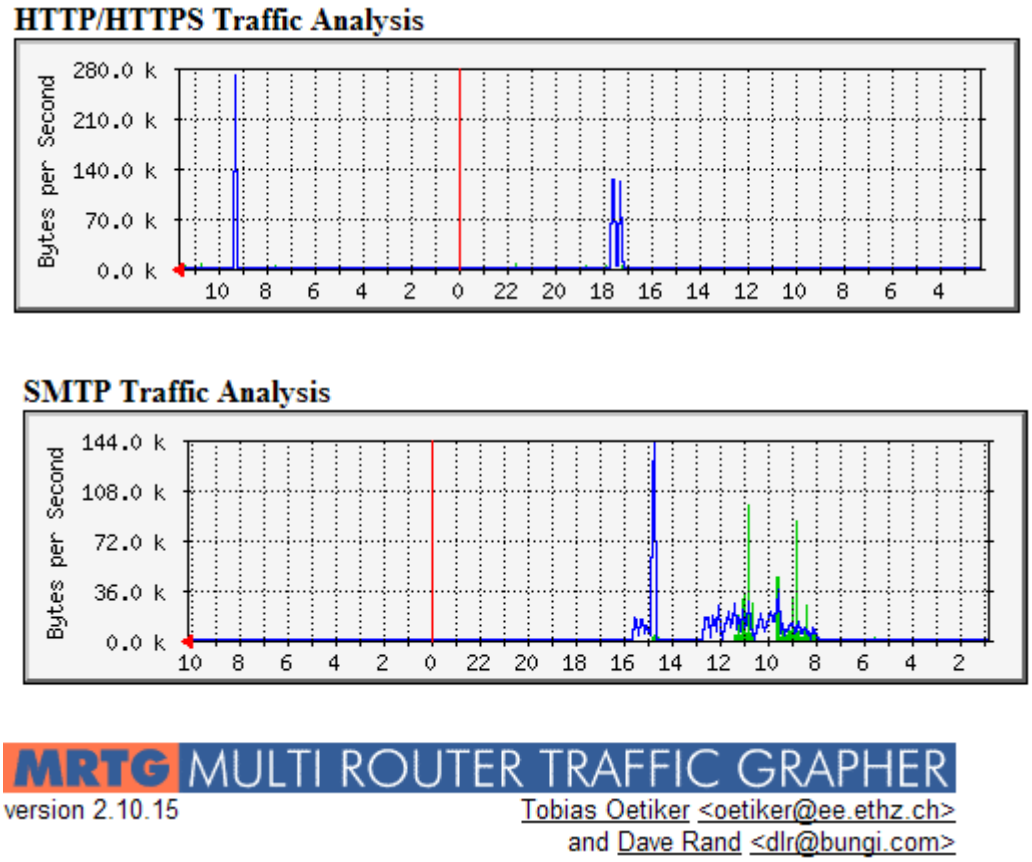


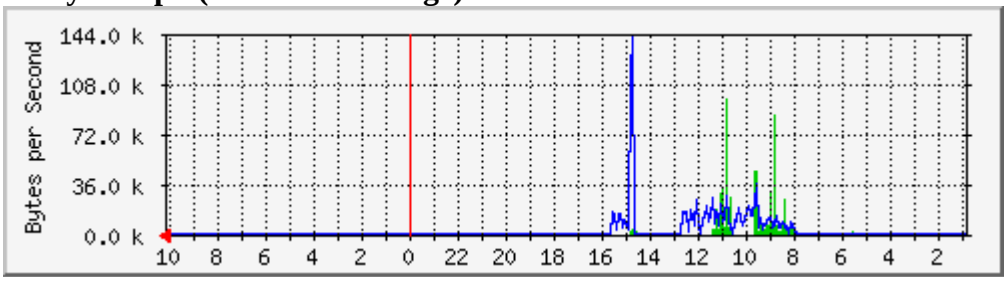
Figure IV-8 : Graphes MRTG pour les trafics relatifs aux services Web et SMTP

On développant le graphe de SMTP par exemple, on accède à une page plus détaillée (cf. figure (IV-9)

SMTP Traffic Analysis

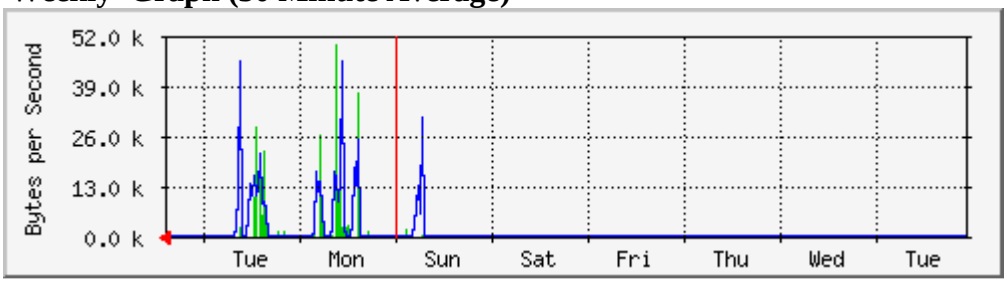
The statistics were last updated **Wednesday, 24 June 2009 at 10:10**,
at which time **'pfp.enp.edu.dz'** had been up for **2 days, 23:14**.

`Daily' Graph (5 Minute Average)



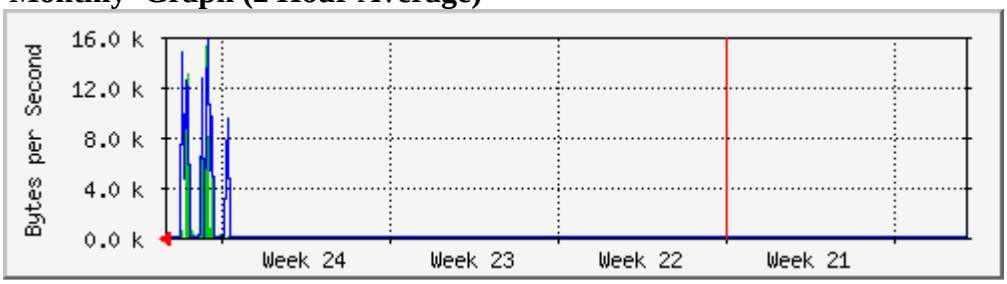
Max In:99.0 kB/s (0.8%) Average In:1545.0 B/s (0.0%) Current In:24.0 B/s (0.0%)
Max Out:143.3 kB/s (1.1%) Average Out:2674.0 B/s (0.0%) Current Out:20.0 B/s (0.0%)

`Weekly' Graph (30 Minute Average)



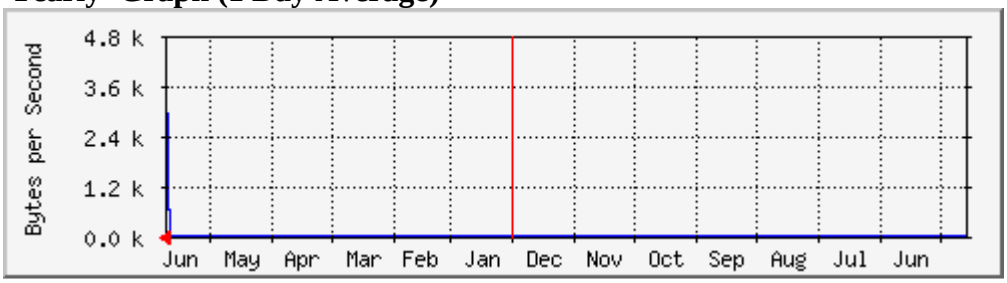
Max In:50.4 kB/s (0.4%) Average In:2180.0 B/s (0.0%) Current In:78.0 B/s (0.0%)
Max Out:45.3 kB/s (0.4%) Average Out:3548.0 B/s (0.0%) Current Out:14.0 B/s (0.0%)

`Monthly' Graph (2 Hour Average)



Max In:15.4 kB/s (0.1%) Average In:2211.0 B/s (0.0%) Current In:23.0 B/s (0.0%)
Max Out:15.9 kB/s (0.1%) Average Out:3603.0 B/s (0.0%) Current Out:8.0 B/s (0.0%)

`Yearly' Graph (1 Day Average)



Max In:3654.0 B/s (0.0%) Average In:1891.0 B/s (0.0%) Current In:3654.0 B/s (0.0%)
Max Out:4618.0 B/s (0.0%) Average Out:2967.0 B/s (0.0%) Current Out:4618.0 B/s (0.0%)

GREEN ### Incoming Traffic in Bytes per Second

BLUE ### Outgoing Traffic in Bytes per Second

Figure IV-9 : Graphe MRTG détaillé pour le trafic relatif au service SMTP

On remarque ici que le trafic sortant est celui en bleu et l'entrant est en vert. En plus des courbes journalières, hebdomadaires, mensuelles et annuelles habituelles, on remarque que des valeurs de débits sont affichées notamment : les valeurs max, actuelles et les valeurs moyennes.

Conclusion

Notre plate-forme de monitoring a été réalisée avec succès. Les services qui y sont monitorés sont réduits mais importants pour l'étude du comportement d'un réseau et le type de trafic qui y circule.

Le service de monitoring par port, à travers la solution que nous avons proposé, nécessite une installation locale de MRTG sur la passerelle en plus d'un serveur APACHE. L'ajout de ces fonctionnalités peut constituer des risques d'insécurité sur cette passerelle.

Il est donc impératif de protéger cette dernière en définissant des règles d'accès adéquates (la passerelle ne doit autoriser que les requêtes HTTP en provenance du serveur de monitoring).

Au final, cette plate-forme n'est qu'un outil qu'il faut placer entre les mains des administrateurs de ce réseau. Ce n'est qu'en arrivant à définir le comportement correct, propre à chaque réseau, que les anomalies peuvent être détectées par la suite sur celui-ci.

C'est alors que les valeurs indiquées par MRTG prendront une réelle signification.

Conclusion générale

Après trois mois de travail, on a pu atteindre notre objectif qui est la mise en œuvre d'une plate-forme de métrologie passive, basée sur les mesures via le protocole SNMP et le Monitoring via l'outil MRTG, pour un réseau de campus universitaire. Ceci nous a permis d'administrer les principaux équipements de ce dernier afin d'en connaitre le comportement plus tard.

Les mesures via SNMP constituent un moyen très rapide et efficace pour la récolte des informations sur un réseau. La limite de ce protocole est fixée par la richesse de la MIB propre aux machines sur lesquels sont installé les Agents. On pensera éventuellement à enrichir cette base d'informations d'où la notion d'extension de la MIB.

MRTG est très facile d'utilisation mais possède quelques défauts qui seront corrigées avec les nouvelles versions compatibles avec RRDTools.

Les mesures passives et actives sont complémentaires et doivent être utilisées simultanément afin d'optimiser la supervision d'un réseau. De ce fait, on pourra parler de la plate-forme de métrologie idéale.

Cette plate-forme doit remplir quelques critères de choix pour sa mise en œuvre, notamment sa compatibilité native avec les éléments actifs, ainsi que son adaptation avec la configuration du réseau administré.

Il convient donc aux techniciens et ingénieurs de ne pas sous-utiliser une plate-forme dont les performances dépassent leurs besoins, et de bien réfléchir au facteur coût de mise en œuvre pour son implémentation.

Toutefois, des outils libres avancés, comme NAGIOS (basé sur SNMP) et CENTREON (interface de monitoring qui exploite NAGIOS), existent et constituent une bonne alternative pour la mise œuvre des systèmes de supervision. Néanmoins, ils nécessitent impérativement la maitrise des procédures d'installation et de configuration, aussi bien des Agents que des Managers.

Annexe A

Présentation du protocole TCP

1. Principe

TCP (qui signifie *Transmission Control Protocol*, soit en français: *Protocole de Contrôle de Transmission*) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle OSI (c'est-à-dire le protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP. TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes :

- TCP permet de remettre en ordre les datagrammes en provenance du protocole IP.
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau.
- TCP permet de formater les données en segments de longueur variable afin de les "remettre" au protocole IP.
- TCP permet de multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne.
- TCP permet enfin l'initialisation et la fin d'une communication de manière courtoise.

2. En-tête TCP

<table>
<tr><td colspan="32">0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</td></tr>
<tr><td colspan="16">Port Source</td><td colspan="16">Port destination</td></tr>
<tr><td colspan="32">Numéro d'ordre</td></tr>
<tr><td colspan="32">Numéro d'accusé de réception</td></tr>
<tr><td colspan="4">Décalagedonnées</td><td colspan="6">réservée</td><td colspan="2">URG</td><td colspan="2">ACK</td><td colspan="2">PSH</td><td colspan="2">RST</td><td colspan="2">SYN</td><td colspan="2">FIN</td><td colspan="16">Fenêtre</td></tr>
<tr><td colspan="16">Somme de contrôle</td><td colspan="16">Pointeur d'urgence</td></tr>
<tr><td colspan="24">Options</td><td colspan="8">Remplissage</td></tr>
<tr><td colspan="32">Données</td></tr>
</table>

Figure A-1 : l'en-tête TCP

- **Port Source** (16 bits): Port relatif à l'application en cours sur la machine source.
- **Port Destination** (16 bits): Port relatif à l'application en cours sur la machine de destination.
- **Numéro d'ordre** (32 bits): indique le numéro du premier octet à envoyer.
- **Numéro d'accusé de réception** (32 bits): Le numéro d'accusé de réception également appelé numéro d'acquittement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu.
- **Décalage des données** (4 bits): il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable
- **Réservé** (6 bits): Champ inutilisé actuellement mais prévu pour l'avenir
- **Drapeaux (flags)** (6x1 bit): Les drapeaux représentent des informations supplémentaires :
    - **URG**: si ce drapeau est à 1 le paquet doit être traité de façon urgente.
    - **ACK**: si ce drapeau est à 1 le paquet est un accusé de réception.
    - **PSH** (PUSH): si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH c'est-à-dire suivant la fonction de livraison des données sans attendre le remplissage des tampons
    - **RST**: si ce drapeau est à 1, la connexion est réinitialisée.
    - **SYN**: Le Flag TCP SYN indique une demande d'établissement de connexion.
    - **FIN**: si ce drapeau est à 1 la connexion s'interrompt.
- **Fenêtre** (16 bits): Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- **Somme de contrôle** (Checksum ou CRC): La somme de contrôle est réalisée en faisant la somme des champs de données de l'en-tête, afin de pouvoir vérifier l'intégrité de l'en-tête
- **Pointeur d'urgence** (16 bits): Indique le numéro d'ordre à partir duquel l'information devient urgente
- **Options** (Taille variable): Des options diverses
- **Remplissage**: On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits.

3. L'encapsulation TCP

Seule la couche supérieure (Application) contient les données et uniquement les données à émettre ou reçues. Chaque couche ajoute ses propres entêtes, encapsulant les paquets de données dans de plus grands paquets, ou enlevant les entêtes dans le cas d'une réception.

Le paquet de données demande à être émis par une application, ces données vont donc recevoir plusieurs entêtes fonction des protocoles utilisés. Dans le cas d'une réception, chaque couche prendra les informations nécessaires et retirera ensuite ses entêtes pour donner le bloc de données restant à la couche de niveau immédiatement supérieur.

Annexe B

Configuration de MRTG sous Windows

SNMP, Perl et MRTG étant installés (voir chapitre III,§4.1.2), on peut désormais entamer la phase de configuration sur la machine Windows locale.

Pour cela, il suffit d'ouvrir une **invite de commande** puis de se déplacer dans le répertoire des fichiers binaires de MRTG à savoir *C:\MRTG\Bin*.

1. Création du fichier de configuration

La construction d'une configuration se fait à l'aide de la commande *cfgmaker* dont le résultat est un fichier « .cfg » comportant les informations nécessaires au suivi du trafic Ethernet sur l'interface administrée.

Exemple : description d'une interface Ethernet dont l'adresse IP est « 192.168.0.2 »

La ligne de commande est la suivante :

```
C:\MRTG\Bin > perl cfgmaker public@192.168.0.2 --output mrtg.cfg
```

L'option « output » spécifie le nom et le chemin du fichier de sortie qui est dans ce cas « mrtg.cfg »

Les paramètres importants à noter, dans mrtg.cfg, sont les suivantes :

```
# WorkDir: c:\mrtgdata
### Interface 16777219 >> Descr: '3Com 3C90x Ethernet Adapter ' | Name: '' | Ip:
'192.168.0.2' | Eth: '00-60-08-4a-37-b2' ###
Target[192.168.0.2_16777219]: 16777219:public@192.168.0.2:
SetEnv[192.168.0.2_16777219]: MRTG_INT_IP="192.168.0.2"
MRTG_INT_DESCR="3Com
3C90x Ethernet Adapter "
MaxBytes[192.168.0.2_16777219]: 12500000
Title[192.168.0.2_16777219]: Traffic Analysis for 16777219 -- BEEPBEEP
PageTop[192.168.0.2_16777219]: <H1>Traffic Analysis for 16777219 -- BEEPBEEP</H1>
```

L'option « WorkDir » est commentée (#) ce qui veut dire que le fichier mrtg.cfg est sous *C:\MRTG\Bin*. Mais si on veut changer son emplacement, il faut agir sur cette ligne en enlevant le « # » puis en spécifiant le chemin au niveau de cette dernière.

Pour ce qui est des autres paramètres :

- Target : La cible qui fera l'objet d'un graphe. Ici la carte réseau dont l'identifiant est 16777219:public@192.168.0.2
- MaxBytes : La valeur maximale pouvant être affichée sur le graphe. Ici 12500000 car il s'agit d'une carte réseau 100 Mbps (soit 12500000 Bytes par seconde).
- Title : MRTG en plus des graphes génèrent des pages HTML pour présenter ces graphes. Cette commande permettra de donner un titre à ces pages Web.
- PageTop : Toujours concernant les pages HTML générées, le code HTML qui se trouve en dessous de cette commande sera inséré en haut de la page.

2. Planification de la génération des graphes

Cette étape consiste à planifier la mise à jour périodique des graphes générés par MRTG.
Pour cela il faut :

- Créer un fichier « mrtg.cmd » avec l'invite de commande puis l'enregistrer sous *C:\MRTG\Bin.*

  Ce fichier doit contenir la ligne suivante :

  ```
  perl C:\MRTG\Bin\mrtg C:\MRTG\Bin\mrtg.cfg
  ```

  qui permet de lancer le programme *C:\MRTG\Bin\mrtg* à l'aide du programme *perl* en lui indiquant d'utiliser le fichier de configuration *C:\MRTG\Bin\mrtg.cfg ;*
- Utiliser le **planificateur de tache** Windows sous **Panneau de configuration** > **Tâches planifiées** afin de créer une nouvelle tache qui exécutera le fichier « mrtg.cfg » toute les 5min.

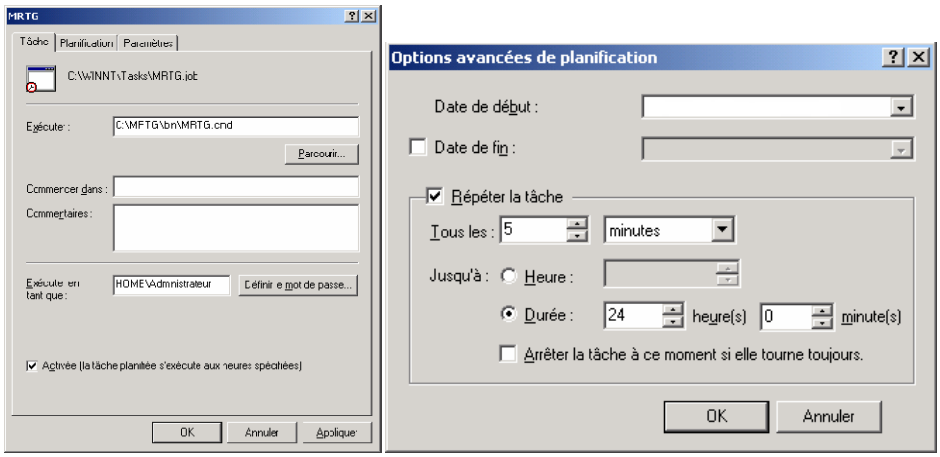


Figure B-1 : Plannification de MRTG sous Windows

Après configuration, il se demander d'allouer cette tache à un compte utilisateur. Il faudra donc désigner un qui possède des droit d'administrateur sur la machine locale afin d'éviter les problèmes de privilèges.

Remarque :

Pour ne pas voir apparaitre la fenêtre d'exécution à chaque mise à jour des graphes, il est préférable d'indiquer un compte qu'on n'utilise pas.

Si la configuration est correcte, de nouveaux fichier doivent apparaitre dans *C:\MRTG\graphes* ;

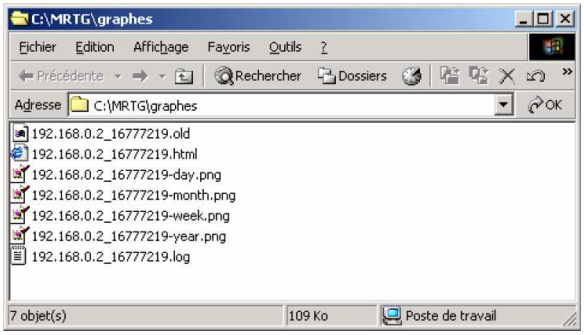


Figure B-2 : Répertoire des graphes de MRTG sous Windows

Ces fichiers étant :

- 192.168.0.2_16777219.log : Les valeurs mesurées par MRTG pour générer le graphe
- 192.168.0.2_16777219.old : Le fichier LOG d'il y a 5 minutes
- 192.168.0.2_16777219-day.png : Le graphique journalier
- 192.168.0.2_16777219-week.png : Le graphique hebdomadaire
- 192.168.0.2_16777219-month.png : Le graphique mensuel
- 192.168.0.2_16777219-year.png : Le graphique annuel
- 192.168.0.2_16777219.html : La page HTML qui présente les graphes

Annexe C

Monitoring des variables SNMP via MRTG

MRTG se base sur SNMP est peut donc être utilisé pour faire le suivi de paramètres, autres que le trafic réseau, à savoir les variables se trouvant dans la MIB de la machine administrée.

Pour ce faire, il suffit juste de spécifier le bon nom du paramètre à visualiser ou d'utiliser l'OID correspondant à ce dernier. Cela, doit figurer au niveau de la ligne « Target » du fichier de configuration pour MRTG.

A noter que :

- L'OID ou le nom utilisé doit être correcte. Une vérification s'impose donc avant d'éditer le fichier « .cfg ».
- Un nom ou OID correcte est celui qui sur lequel on récupère une valeur (obtention d'une réponse) par une requête *snmpget*.

Exemple d'une bonne requête :

```
# snmpget -v 2c -c public localhost system.sysUpTime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (13966) 0:02:19.66
```

Ceci étant dit, on va donner quelques exemples de fichiers de configurations concernant divers paramètres système figurants dans la MIB à savoir : la charge CPU, l'utilisation de la RAM et le nombre de connexions TCP actives sur la machine locale.

1. La charge CPU

Il existe plusieurs approches pour visualiser l'utilisation CPU car le système tient plusieurs compteur à cet effet : Idle, System, User, Nice. Et comme MRTG permet l'affichage de deux courbes à la fois, on pourra faire des comparaisons entre deux paramètres ou voir la somme de nombreux paramètres.

1.1 User vs Idle

```
LoadMIBs: /usr/share/snmp/mibs/UCD-SNMP-MIB.txt

Target[cpu]:ssCpuRawUser.0&ssCpuRawIdle.0:public@localhost
RouterUptime[cpu]: public@localhost
MaxBytes[cpu]: 100
Title[cpu]: CHARGE CPU
PageTop[cpu]: <H1>Charge CPU (User et Idle) %</H1>
ShortLegend[cpu]: %
YLegend[cpu]: Utilisation CPU
Legend1[cpu]: User CPU en % (Charge)
Legend2[cpu]: Idle CPU en % (Charge)
LegendI[cpu]:  User
LegendO[cpu]:  Idle
Options[cpu]: growright,nopercent
```

La ligne LoadMIBs permet ici d'utiliser le nom du paramètre à surveiller au lieu de mettre son OID.

1.2 User vs System

```
LoadMIBs: /usr/share/snmp/mibs/UCD-SNMP-MIB.txt

Target[usrsys]:ssCpuRawUser.0&ssCpuRawSystem.0:public@localhost
RouterUptime[usrsys]: public@localhost
MaxBytes[usrsys]: 100
Title[usrsys]: CPU LOAD
PageTop[usrsys]: <H1> Charge CPU (User et System) %</H1>
ShortLegend[usrsys]: %
YLegend[usrsys]: Utilisation CPU
Legend1[usrsys]: User CPU en % (Charge)
Legend2[usrsys]: System CPU en % (Charge)
LegendI[usrsys]:  User
LegendO[usrsys]:  System
Options[usrsys]: growright,nopercent
```

1.3 Utilisation CPU active

Cet exemple gère la somme des compteurs : User, System et Nice pour les fusionner leurs graphes respectifs en un seul.

```
LoadMIBs: /usr/share/snmp/mibs/UCD-SNMP-MIB.txt

Target[cpusum]:ssCpuRawUser.0&ssCpuRawUser.0:public@localhost +
        ssCpuRawSystem.0&ssCpuRawSystem.0:public@localhost +
        ssCpuRawNice.0&ssCpuRawNice.0:public@localhost

RouterUptime[cpusum]: public@localhost
MaxBytes[cpusum]: 100
Title[cpusum]: CHARGE CPU
PageTop[cpusum]: <H1>Charge CPU Active %</H1>
ShortLegend[cpusum]: %
YLegend[cpusum]: Utilisation CPU
Legend1[cpusum]: CPU Active en % (Charge)
LegendI[cpusum]:  Active
Options[cpusum]: growright,nopercent
```

2. Nombres de connections TCP actives

```
Target[tcpopen]: .1.3.6.1.2.1.6.9.0&.1.3.6.1.2.1.6.9.0:public@localhost

Options[tcpopen]: nopercent,growright,gauge,noinfo
Title[tcpopen]: Open connections TCP Actives
PageTop[tcpopen]: <H1> Open connections TCP Actives </H1>
MaxBytes[tcpopen]: 1000000
YLegend[tcpopen]: # conns
ShortLegend[tcpopen]: connections
LegendI[tcpopen]:  Connections:
Legend1[tcpopen]: connections TCP Actives
```

3. RAM non utilisée

```
Target[freemem]: .1.3.6.1.4.1.2021.4.11.0&.1.3.6.1.4.1.2021.4.11.0:public@localhost

Options[freemem]: nopercent,growright,gauge,noinfo
Title[freemem]: <H1> RAM disponible </H1>
PageTop[freemem]: RAM disponible
MaxBytes[freemem]: 1000000
YLegend[freemem]: bytes
ShortLegend[freemem]: bytes
LegendI[freemem]: RAM disponible
Legend1[freemem]: RAM disponible, swap non incluse, en bytes
```

A

Adresse MAC Adresse « physique » (car intégrée à l'équipement) permettant d'identifier un host (poste client ou serveur) sur un réseau local. L'adresse MAC est codée sur 6 octets. Les trois premiers correspondent au code constructeur (attribué par l'IEEE), les trois derniers sont gérés par le constructeur lui-même.

Agent Elément logiciel embarqué (implémenté) dans un élément actif du réseau permettant sa gestion par une station de supervision.

ARP *Address Resolution Protocol.* Protocole de service qui permet à une machine du réseau IP de connaitre l'adresse MAC de la machine du réseau physique à qui sont destinées les données en instance d'émission. Les spécifications du protocole ARP sont définies dans la RFC 826 de novembre 1982.

ASN.1 *Abstract Syntax Notation n°1*. Langage de description de MIB défini par l'ISO.

B

Bit Elément protocolaire essentiel utilisé en informatique et dans les réseaux et télécommunications. Acronyme de *Binary Digit.* A pour valeur 0 ou 1.

Broadcast Diffusion. Emission, sur un réseau local Ethernet, d'une trame à destination de toutes les machines. Tous les bits du champ d'adresse de destination sont positionnés à 1 (FFFFFF : FFFFFF). Le broadcast est utilisé par des protocoles tels ARP et DHCP.

Byte Equivalent d'un octet, soit 8 bits.

C

Collision Phénomène résultant de l'émission simultanée de deux trames Ethernet, alors que les stations ont considéré le media comme disponible. La collision est traitée par le protocole CSMA/CD. La collision, dans un réseau local Ethernet, n'est pas considérée comme une erreur, mais comme un événement normal (et traité). Le taux maximum admissible est de 1/1000 soit une collision pour mille trames utiles.

Communauté Deux sens bien distincts. Association, dans un réseau IP, d'Agent et d'un et d'un Manager. Mot de passe échangé entre Agent et Manager permettant l'accès aux informations de la MIB.

Commutateur Ethernet Elément actif de réseau, fonctionnant au niveau II du modèle OSI de l'ISO. Le commutateur tient compte de l'en-tête de la trame Ethernet et de ses informations (adresse MAC, attributs de VLAN ou de QoS).

Compilation Dans le sens SNMP, opération d'intégration d'une MIB privée dans une station de supervision. Dans un sen plus commun, transformation d'instruction de langage de développement en langage interprétable par la machine sur lequel va s'exécuter le programme.

Couche Entité d'un modèle de référence. Une couche est un ensemble de fonctions fournissant un service spécifique :

- La couche physique (OSI I) prend en charge l'aspect connectique et signaux électriques sur le support de transmission.
- La couche liaison (OSI II) gère les trames et le contrôle d'erreurs.
- La couche réseau (OSI III) gère l'adressage et le routage.
  Voir modèle en couche.

D

Datagramme Appellation anglo-saxonne d'un bloc de données de niveau OSI III échangé sur un réseau en mode non connecté. Le mode de communication sans connexion est dit en mode datagramme.

DHCP *Dynamic Host Configuration Protocol*. Protocole de configuration automatique des nœuds d'un réseau (hosts) en allouant dynamiquement une adresse IP, le masque et les éventuels paramètres complémentaires (adresse du routeur par défaut, adresse(s) du ou des serveur(s) DNS).

DNS *Domain Name System* : protocole et technologie permettant d'associer nom de domaine ou bien de machine à une adresse IP. Le DNS est notamment utilisé dans le réseau Internet.

E

Ecoute Deux sens. Analyse du protocole sur un support de transmission à l'aide d'un équipement dédié (exemple du logiciel Wireshark embarqué sur un PC) en vue de dépannage ou de mise au point. Détection d'émission sur un réseau Ethernet.

EGP *Exterior Gateway Protocol*. Protocole de routage utilisé dans le réseau Internet. Est défini par la RFC 904, d'avril 1984.

Elément actif Terme générique définissant tout appareil utilisé dans un réseau de données. OSI I (répéteurs, hubs), OSI II (ponts Ethernet, commutateurs ou switches Ethernet, bornes Wi-Fi), OSI III et plus (routeurs, passerelles applicatives).

En-tête Première partie de la trame. L'en-tête comporte, de façon générique, des adresses source et destination, l'identifiant du

protocole transporté et, parfois, une indication de la longueur totale de la trame.

Ethernet Technologie de réseau local sur une technologie logique bus, définie par Digital, Intel et Xerox au début des années 70's. Standardisée 802.3 par l'IEEE. Déclinée sous multiples formes (média et débits). Standard de fait, représentant aujourd'hui plus de 95% des réseaux locaux d'entreprises. Normalisée 8802.3 par l'ISO.

F

Filtrage Un filtre paramétré dans un analyseur de protocole permet de ne capturer que les trames concernées par une application donnée. Le filtre peut s'appliquer à une adresse MAC, une adresse IP, un type de protocole, un numéro de port ou une combinaison des différents critères.

I

ICMP *Internet Control Message Protocol*. Protocole de service qui permet la communication de signalisation entre équipements (ICMP-Redirect) et les tests nécessaires dans un inter-réseau (ping, traceroute). Les messages ICMP sont également émis par des routeurs en cas de problème de routage (destination unreachable) ou de fragmentation. Les spécifications du protocole ICMP sont définies dans la RFC 792 de septembre 1981.

IEEE *Institute of Elecrical and Electronics Engineers*. Organisation professionnelle à but non lucratif, constituée d'ingénieurs

électroniciens, d'informaticiens et de professionnels de télécommunications, ayant pour vocation de promouvoir la connaissance dans le domaine de l'ingénierie électrique.

Interface Point de connexion d'un système ou d'un protocole. Frontière entre deux entités. L'interface série permet, par exemple, de connecter un équipement de traitement de données à un système de transmission de données.

Internet Réseau mondial créé par l'interconnexion des réseaux public et privés et fournissant de multiples services.

IP *Internet Protocol*. Protocole de niveau OSI III, prenant en charge les fonctions d'adressage et de routage dans un réseau maillé. Le protocole IP, fonctionnant en mode non connecté, est défini par la RFC 791 de septembre 1981.

ISO *International Standard Organisation*. Organisation internationale de normalisation dans le siège est basé à Genève. L'ISO a défini un modèle d'architecture d'un système ouvert appelé OSI (*Open System Interconnection*).

L

LAN *Local Area Network*. Réseau local de l'entreprise, de type Ethernet dans 99% des cas. Le réseau local assure une couverture restreinte (quelques centaines de mètres) avec un débit important (le Gigabit Ethernet est très courant de nos jours). Le LAN, à l'échelle d'un bâtiment, est toujours d'exploitation privée.

M

MAIL message électronique. Au sens courant, courrier électronique.

Management **S**upervision/gestion des éléments actifs du réseau et de ses équipements connexes.

Manager Station en charge de la supervision des éléments actifs du réseau.

Message Information échangée entre deux entités. Egalement appelé PDU (*Protocol Data Unit*).

MIB *Management Information Base*. Ensemble des données gérées par l'Agent, implémenté dans l'élément actif, permettant sa gestion par le Manager. La MIB répond aux standards définis par les RFC's 1156 et 1213 (MIB standard).

Mode non connecté Méthode d'échange d'informations sans assurance de l'existence du destinataire, sans gestion d'erreur. Les protocoles IP et UDP fonctionnent en mode non connecté.

Modèle en couche Modèle destiné à l'interopérabilité des systèmes (modèle OSI défini par l'ISO). Les systèmes conformes au modèle sont à même de s'interconnecter, d'échanger les informations, de faire interagir des applications.

N

NVRAM Mémoire vive (RAM) non volatile. L'information est mémorisée dans la NVRAM (même lorsque l'appareil est hors tension). Voir RAM.

O

Octet Ensemble de 8 bits. Egalement appelé byte.

OID *Object Identifier*. Dans une MIB, identifiant d'une variable par un code numérique unique.

Opérateur Fournisseur de service de transmission de données via Internet ou de service de téléphonie (fixe ou mobile).

OSI *Open System Interconnection.* Modèle d'architecture de système ouvert, défini par l'ISO au début des années 80's. La motivation était l'interopérabilité des systèmes, l'interaction des applications et l'échange des informations.

P

Paquet Ensemble de données protocolaires de niveau OSI III. Appelé dans un environnement TCP/IP.

PDU *Protocol Data Unit*. Terme générique définissant un bloc de données protocolaires de niveau N. Dans la littérature OSI, on parlera de Data-Link-PDU (trame), de Network-PDU (paquet), de Transport-PDU …

Port Interface entre protocole de transmission (UDP ou TCP) et application. Le port permet l'adressage des données applicatives sur une connexion de transport multiplexée.

Protocole Ensemble exhaustif de règles définissant les échanges des données entre deux équipements. On parle de protocole de niveau N. Les protocoles font l'objet de standards RFC dans le cas de l'IP, de normes ISO dans le cas d'une implémentation OSI.

Q

QoS — *Quality of Service*. La qualité de service est la capacité à véhiculer dans de bonnes conditions un type de trafic donné, en terme de disponibilité, débit, délai de transit et taux de perte de paquets. La QoS est nécessaire à certaines applications en temps réel (téléphonie notamment).

R

RAM — *Random Access Memory*. Mémoire vive d'un système informatique dans laquelle est chargé le système d'exploitation et la configuration.

Requête — Interrogation dans une base de données, en vue d'exploiter l'information obtenue.

RFC — *Request For Comments.* Document produit par l'IETF, définissant le fonctionnement d'un protocole (IP) ou d'un service (DNS). Les RFC's sont disponibles au format texte sur le site *http://www.ietf.org*

Routeur — Elément d'interconnexion de niveau OSI III permettant l'interconnexion des réseaux de technologies hétérogènes. Les routeurs échangent des datagrammes.

S

Serveur — Machine assurant la mise à disposition des données dans un réseau informatique, ou des services particuliers (application, sauvegarde, impression …)

Service Fonctionnalité apportée par un serveur ou un équipement dédié du réseau informatique. Dans le sens OSI, une couche fournit des services (contrôle de flux, gestion d'erreur …)

SMTP *Simple Mail Transfer Protocol.* Protocole de messagerie utilisé sur Internet. Le protocole SMTP utilise les services en mode connecté de TCP. SMTP est défini par les RFC's 788 de novembre 1981 et 2821 d'avril 2001.

Sniffer Logiciel spécifique embarqué, dans la plupart des cas, sur un ordinateur portable destiné à l'analyse en temps réel des données protocolaires échangées sur un réseau local ou un réseau d'interconnexion. Utilisé en dépannage et pour des audits.

SNMP *Simple Network Management Protocol*. Protocole de supervision de réseau. Protocole d'échange d'informations entre Agent et Manager. Les spécifications du protocole SNMP sont définies dans la RFC 1157 de mai 1990.

Station Dans un réseau informatique, poste client. Le terme station s'applique aussi à une plate-forme de supervision.

Superviseur Equipement assurant les services de supervision.

Supervision Surveillance de l'état d'un réseau et de ses composants. Par extension, s'applique à tous les équipements d'un système informatique.

Switch Commutateur.

Système d'exploitation Ensemble de logiciels permettant le fonctionnement d'une machine informatique. Le système d'exploitation prend en charge les périphériques de stockage, d'impression et d'entré/sortie. Les communications entre les ordinateurs sont dans la plupart des cas assumées par le système d'exploitation.

T

TCP *Transmission Control Protocol*. Protocole de transport utilisé dans les réseaux IP, en mode connecté, assurant un service fiable entre applications. Les spécifications du protocole TCP sont définies dans la RFC 793 de septembre 1981.

TELNET Protocole d'émulation de terminal asynchrone sur un réseau IP, permettant la connexion à distance en mode écran-clavier. Les dernières spécifications du protocole TELNET sont définies dans la RFC 854 de mai 1983.

Terminal Appareil de communication simpliste (sans ressources de calcul ou de stockage de données) composé d'un écran et d'un clavier. Fonctionne, dans la plupart des cas, selon un mode asynchrone.

Trame Suite de bits formalisée, composé d'un en-tête, d'un champ de données, et d'un certificat calculé selon un algorithme défini. La trame est dans la plupart des cas, équivalente à un bloc de données de niveau OSI II.

Trap Message d'alarme émis spontanément par un Agent vers son Manager en cas d'incident ou de changement d'état d'une interface.

U

UDP *User Datagram Protocol.* Protocole de transmission en mode non connecté. Les spécifications du protocole UDP sont définies dans la RFC 768 d'août 1980.

V

Variable Information contenue dans la MIB d'un élément actif (compteur d'erreurs, Uptime, entrée dans la table de routage), exploité par la Manager. Dans ce cas, une variable peut être de type numérique ou de type texte.

Vitesse Rapidité de transmission d'un réseau (bande passante). Correspond au débit des éléments protocolaires sur un support de transmission. Le débit est exprimé en nombre de bits par secondes.

VLAN *Virtual Local Area Network.* Segmentation (découpage) d'un réseau local Ethernet, ayant pour objectif :

- d'optimiser les performances en réduisant les collisions.
- d'améliorer la sécurité en isolant les flux de façon étanche.

Les VLAN's peuvent être construits en tenant compte des adresses MAC des machines, des ports sur lesquels elles sont connectées, de l'adresse IP du sous-réseau.

Webgraphie/Bibliographie

[I.1] http://www-sop.inria.fr/rodeo/avega/phd/phd-html/node51.html

[I.2] Réseaux informatiques notions fondamentales, Philippe Antelin-José Dordoigne.

[I.3] François PIGNETS. *Réseaux Informatiques: Supervision et Administration* . (2007)
Editions ENI- http://www.editions-eni.com

[I.4] Philippe OWEZARSKI, *CNRS* LAAS *et* Nicolas LARRIEU. *Techniques et outils de métrologie pour l'Internet et son trafic*. Technique de l'ingénieur *R1090.*

[I.5] Endace: http://www.endace.com/

[I.6] Ipanema Technologies http://www.Ipanematech.com/

[I.7] Qosmetrics http://www.qosmetrics.net/

[I.8] PAXSON (V.), ALMES (G.), MAHDAVI (J.) et MATHIS (M.). *Framework for IP performance metrics*. RFC 2330 May (1998).

[I.9] ALMES (G.), KALIDINDI (S.) et ZEKAUSKAS (M.). – *A one-way delay metric for IPPM*. RFC 2679 September (1999).

[I.10] ALMES (G.), KALIDINDI (S.) et ZEKAUSKAS (M.). – *A one-way packet loss metric for IPPM*. RFC 2680 September (1999).

[I.11] ALMES (G.), KALIDINDI (S.) et ZEKAUSKAS (M.). – *A round-trip delay metric for IPPM*. RFC 2681 September (1999).

[II.1] Géraldine TEXIER et Octavio MEDINA. *Métrologie des réseaux*. Technique de l'ingénieur te7605.

[II.2] RFC 1157 – Simple Network Management
Protocol (SNMP) http://www.ietf.org/

[II.3] Douglas Mauro, Kevin Schmidt. *Essential SNMP, 2nd Edition.* O'Reilly September 2005

Bibliographie

[II.4] James Kretchmar. *Open Source Network Administration.* Prentice Hall PTR September 22 (2003)

[II.5] Pan, Heng. *SNMP-based ATM Network Management*. Artech House, Inc. (1998)

[II.6] White paper. Netflow services and applications. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/

[II.7-10] Olivier WILLM. *Administration de réseaux informatiques : protocole SNMP*.

Technique de l'ingénieur h2840.

[II.8] document (article) NET-SNMP: L'ADMINISTRATION DE RESEAU par Patrice KADIONIK, Maître de Conférence à l'ENSEIRB

[II.9] http://christian.caleca.free.fr/snmp/principe.htm

[II.11] les oid sous private.enterprise

http://www.iana.org/assignments/enterprise-numbers

[III.1] Vittoria Rezzonico & Appoline Raposo de Barbosa. *Le monitoring avec Nagios.* mars 2008

http://ditwww.epfl.ch/SIC/SA/SPIP/Publications/spip.php?article1450

[III.2] site de Tobi Oetiker's MRTG - The Multi Router Traffic Grapher

http://oss.oetiker.ch/mrtg/

[III.3] Thomas Boutell. GD Lib, a graphics library for fast creation of GIF images. http://www.boutell.com/gd/

[III.4] Tobias Oetiker – Swiss Federal Institute of Technology, Zurich. *MRTG – The Multi Router Traffic Grapher*

الملخص

أخذت الشبكات المعلوماتية تتوسع باستمرار في السنوات الأخيرة. مع هذا التوسع ظهرت خدمات جديدة (الهاتف عبر بروتوكول الانترنيت، الفيديوكونفيرنس، برامج الشاشة على الانترنيت ...) مما يوجب احترام و توفير حد مقبول فيما يتعلق بنوعية هذه الخدمات.

ظهر القياس كحل لهذه المتطلبات، الأمر الذي يمكن المسيرين من معرفة و فهم سلوك شبكاتهم المعلوماتية. بهذا النحو يصبح التدخل السريع، في حالة وجود مشاكل، سهلا مع إمكانية إضافة وظائف جديدة على مستوى هذه الشبكات.

يتمثل مشروعنا في استعمال خصائص القياس السلبي حتى نتمكن من تنصيب منبر إشراف على مستوى الشبكة لحرم جامعي. هذه الخدمة ستمنح مدير هذه الشبكة فرصة تتبع سلوكها قصد إجراء التعديلات اللازمة عليها و بالتالي تحسين أدائها.

Résumé

De nos jours, les réseaux informatiques ne cessent de s'étendre. De nouveaux services font leur apparition (téléphonie IP, visioconférence, TVOIP ….) et une certaine qualité de service doit être entretenue.

La métrologie vient répondre à ces besoins, en permettant aux administrateurs de connaitre et de comprendre le comportement de leurs réseaux. Ainsi, intervenir rapidement en cas de problème devient plus facile, et la planification pour l'ajout de nouvelles fonctionnalités sera envisageable.

Notre projet consiste à utiliser l'approche passive de la métrologie afin de mettre en œuvre une plate-forme de supervision pour le réseau d'un campus universitaire. Ce service permettra, au gestionnaire de ce réseau, de suivre le comportement de celui-ci afin d'y apporter les modifications adéquates et donc, en améliorer les performances.

Abstract

Nowadays, networks are constantly expanding. New services are emerging (IP telephony, videoconferencing, TVOIP…) and a certain quality of service must be maintained.

Metrology is a respond to these needs by enabling administrators to know and understand the behavior of their networks. Thus, to intervene if a problem occurs, becomes easier, and planning to add new features will be possible.

Our project is to use the passive approche of Metrology in order to implement a monitoring-platform for the network of a university campus. This service gives the this network's operator the opportunity to check its behavior, and apply the appropriate changes on it in order to improve its performance.

MOT CLEFS/ KEY WORDS : SNMP, MIB, MRTG, Métrologie, Monitoring