

---

# Module M3108c

# Supervision Réseau

---

DUT R&T 2ème année  
Laurent Gallon

---

# PLAN

---

## 1 - Introduction

2 - Cartographie de présence

3 - Supervision basée sur SNMP

4 - Supervision basée sur SYSLOG

5 - Quelques logiciels de supervision

---

# 1 - Introduction

---

- L'administrateur réseau a plusieurs rôles, et en particulier :
    - vérifier que ses matériels fonctionnent bien
    - surveiller l'évolution du trafic sur son réseau
  - On dit que l'administrateur **supervise** son réseau = “surveille” son réseau
-

# 1 - Introduction

---

- Superviser un réseau, c'est :
    - mieux connaître son état et son comportement
      - notamment le “profil” du trafic
    - contrôler son bon dimensionnement
      - y a-t-il des liens saturés (collisions, ...) ?
    - détecter les pannes éventuelles
      - matérielles ou logicielles
    - détecter les intrusions / attaques
      - opérations ou profil réseau anormaux
-

# 1 - Introduction

---

- Que peut-on / doit-on superviser (surveiller)?
    - Les matériels actifs :
      - switches, routeurs, bornes wifi, accès internet, ...
    - Les serveurs :
      - authentification, stockage, annuaires, web, autocommutateurs tél., ...
    - Postes clients :
      - PCs, portables, postes tél., smartphones, tablettes, ...
-

# 1 - Introduction

---

- Les différents moyens pour superviser
    - test de présence des matériels et des services
      - ping, connexion TCP, ...
    - récolte de valeurs de variables sur les matériels
      - MIB + SNMP
    - récolte des journaux systèmes sur les matériels
      - SYSLOG
  - La plupart des logiciels de supervision intègrent ces trois méthodes
-

# PLAN

---

1 - Introduction

**2 - Cartographie de présence**

3 - Supervision basée sur SNMP

4 - Supervision basée sur SYSLOG

5 - Quelques logiciels de supervision

---

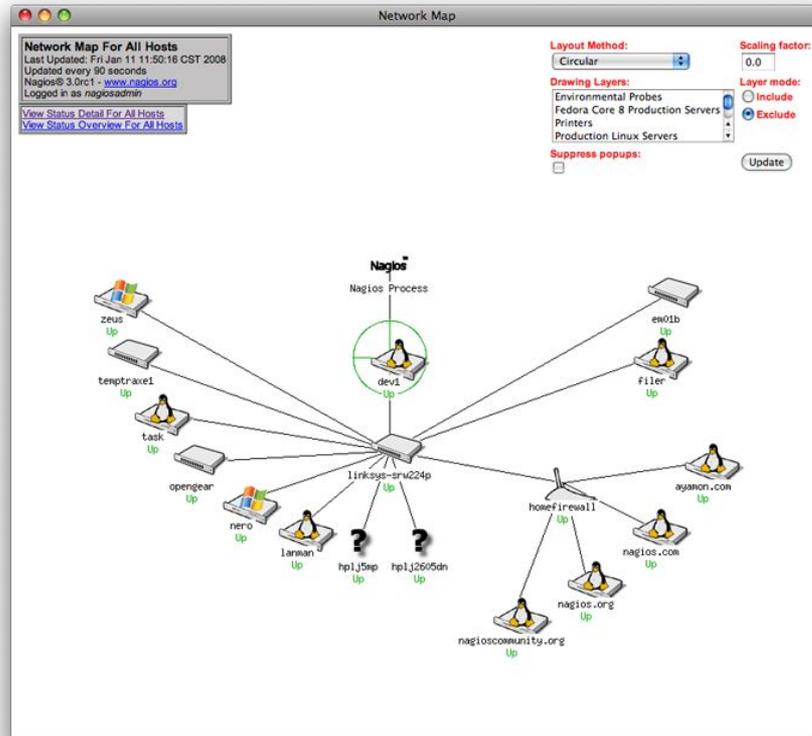
## 2 - Cartographie de présence

---

- C'est le premier niveau de supervision
  - On cherche à tester la présence des matériels, et des services sur les matériels
  - Objectif
    - Créer une cartographie, avec un code de couleur
      - vert = le matériel / service est présent
      - jaune / orange = ne répond pas aux dernières demandes
      - rouge = ne répond pas depuis longtemps
-

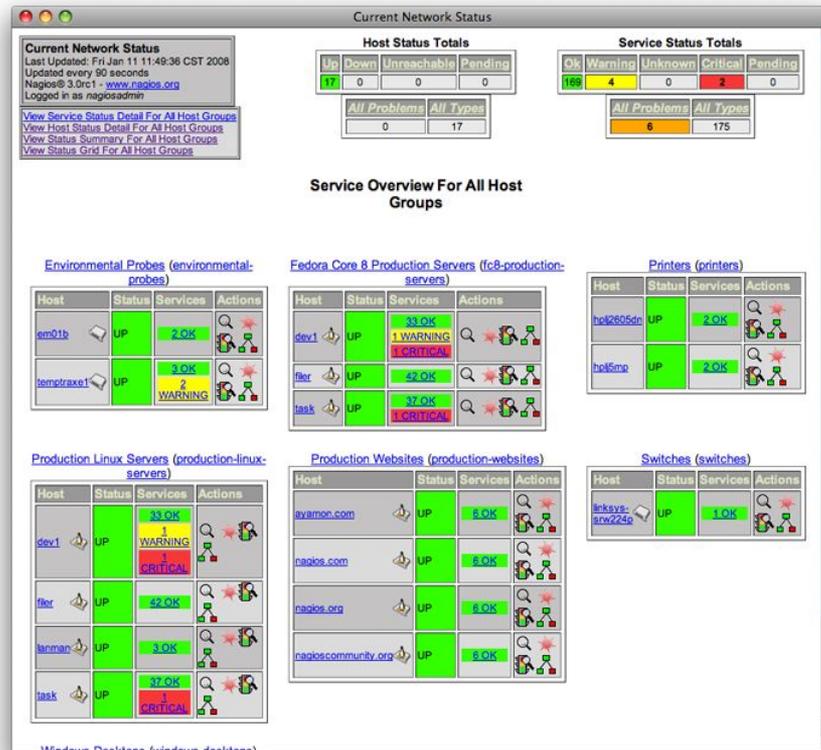
# 2 - Cartographie de présence

Exemple issu du site officiel de Nagios



# 2 - Cartographie de présence

Exemple issu du site officiel de Nagios



# 2 - Cartographie de présence

---

- Présence “physique” des matériels
    - un simple “ping ip” sur le matériel
  
  - Présence des services sur un matériel
    - un test d’ouverture de connexion pour les services basés sur TCP
      - ex : web(80/8080), ftp(21), telnet(23), ssh(22)
    - un “ping UDP” pour les services basés sur UDP
      - ex : DNS(53), DHCP(67), ...
-

## 2 - Cartographie de présence

---

- Les tests de connexion sont effectués régulièrement
    - “polling” : on relance le test toutes les X secondes
  - On configure le nbre d’essais infructueux successifs qui font passer d’une couleur à une autre
    - ex : vert à jaune = 1 ; jaune à orange = 5; orange à rouge = 10
-

# PLAN

---

- 1 - Introduction
  - 2 - Cartographie de présence
  - 3 - Supervision basée sur SNMP**
  - 4 - Supervision basée sur SYSLOG
  - 5 - Quelques logiciels de supervision
-

# 3 - Supervision basée sur SNMP

---

## 3.1 PRINCIPES DE BASE

- SNMP ?
    - Simple Network Management Protocol
    - Protocole de communication permettant à deux entités réseaux d'échanger des infos sur leur état
-

# 3 - Supervision basée sur SNMP

---

- Objectifs de la supervision SNMP :
    - récupérer les valeurs de métriques sur une machine distante (=agent)
    - monitorer (=suivre l'évolution) de certaines métriques
    - éventuellement modifier les valeurs de certaines variables sur l'agent distant
    - être informé en temps réel d'événements importants ayant lieu sur l'agent (=trap)
-

# 3 - Supervision basée sur SNMP

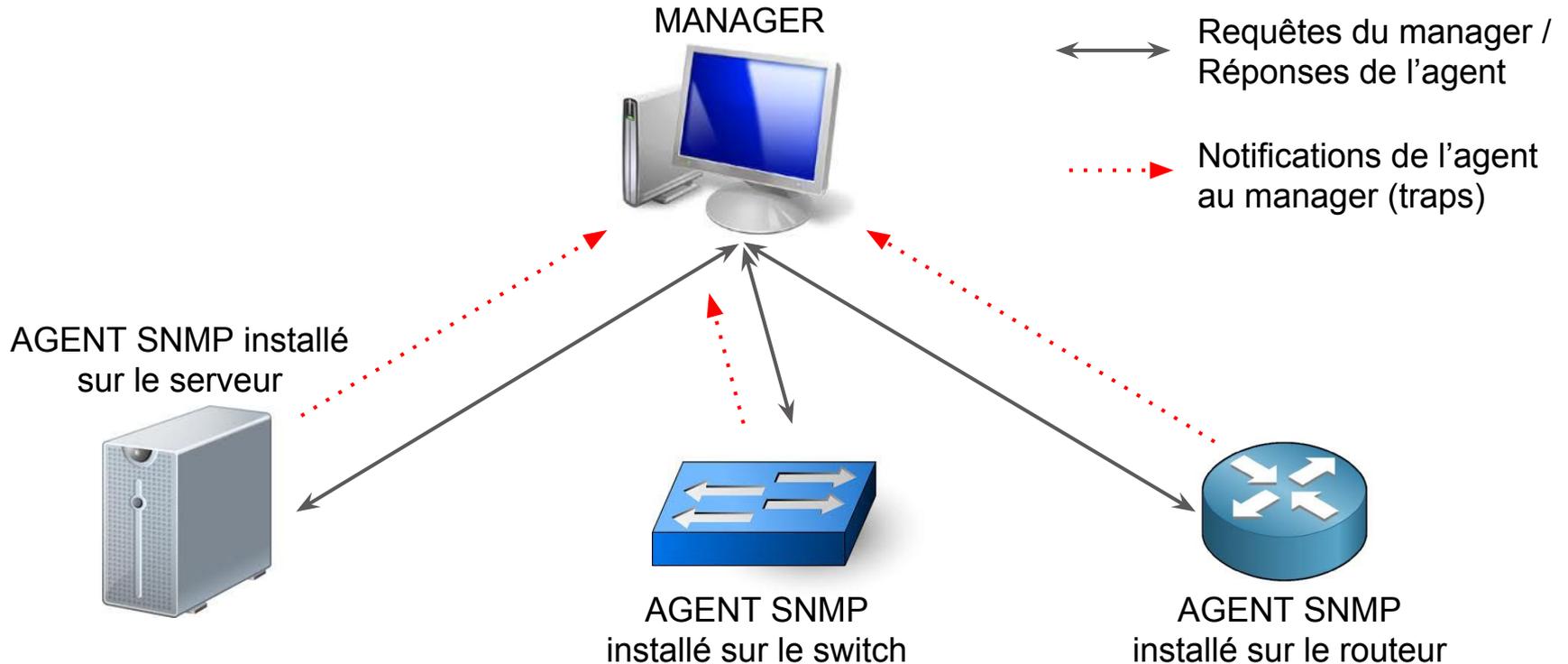
---

## 3.2 MODELE AGENT - MANAGER

- **Manager**
    - entité qui supervise le réseau
  - **Agent**
    - programme installé sur un matériel réseau / serveur
    - répond aux sollicitation du manager pour lui donner son état
    - informe le manager d'événements exceptionnels
-

# 3 - Supervision basée sur SNMP

---



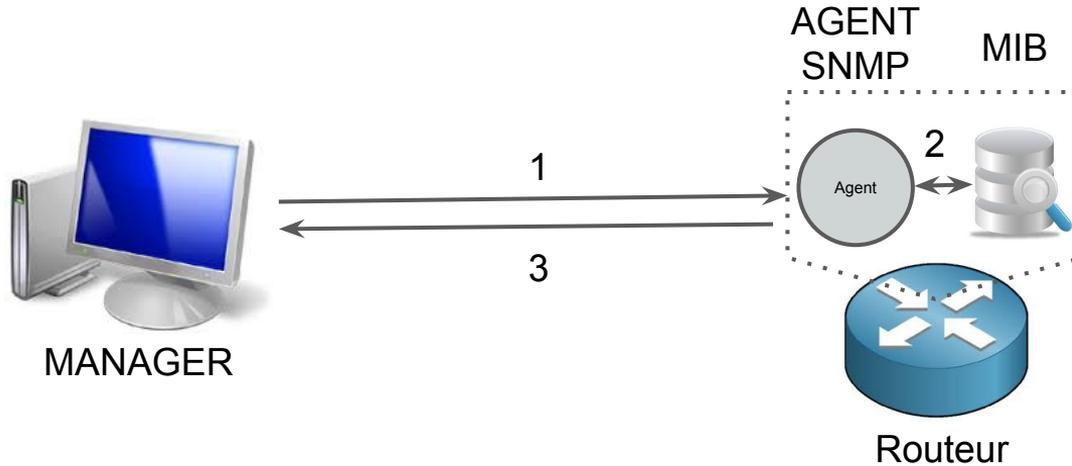
# 3 - Supervision basée sur SNMP

---

- Que demande le manager à l'agent ?
    - récupération des valeurs de variables
      - texte, valeurs numériques, dates, ...
    - variables contenues dans une base de données
      - MIB : Management Information Base
-

# 3 - Supervision basée sur SNMP

---



- 1 : la manager demande à l'agent de récupérer la valeur d'une variable
  - 2 : l'agent réceptionne la demande, et questionne sa MIB (recherche valeur)
  - 3 : l'agent renvoie la valeur au manager
-

# 3 - Supervision basée sur SNMP

---

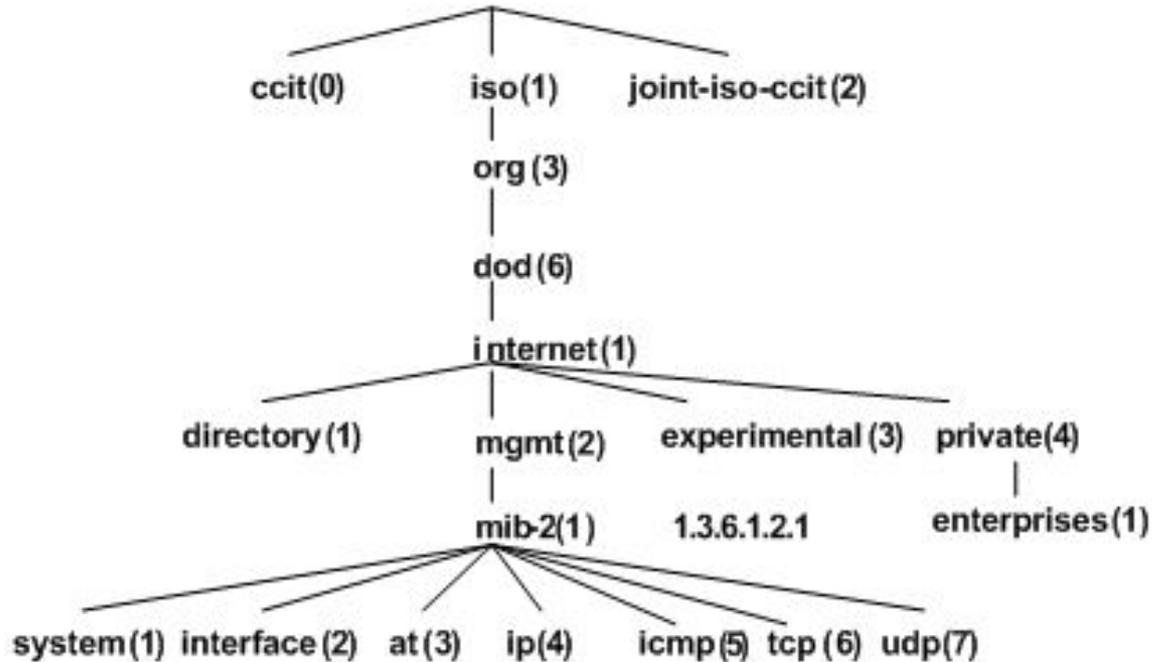
## 3.3 MIB

### 3.3.1 - Généralités

- MIB = Management Information Base
  - Système d'information (base de données) situé sur chaque matériel géré par un agent SNMP
  - Structure de la MIB = arbre
    - les variables = feuilles de l'arbre
    - chaque noeud identifié par un numéro et un nom symbolique (OID)
-

# 3 - Supervision basée sur SNMP

---



# 3 - Supervision basée sur SNMP

---

## 3.3.2 - OID (Object Identifier)

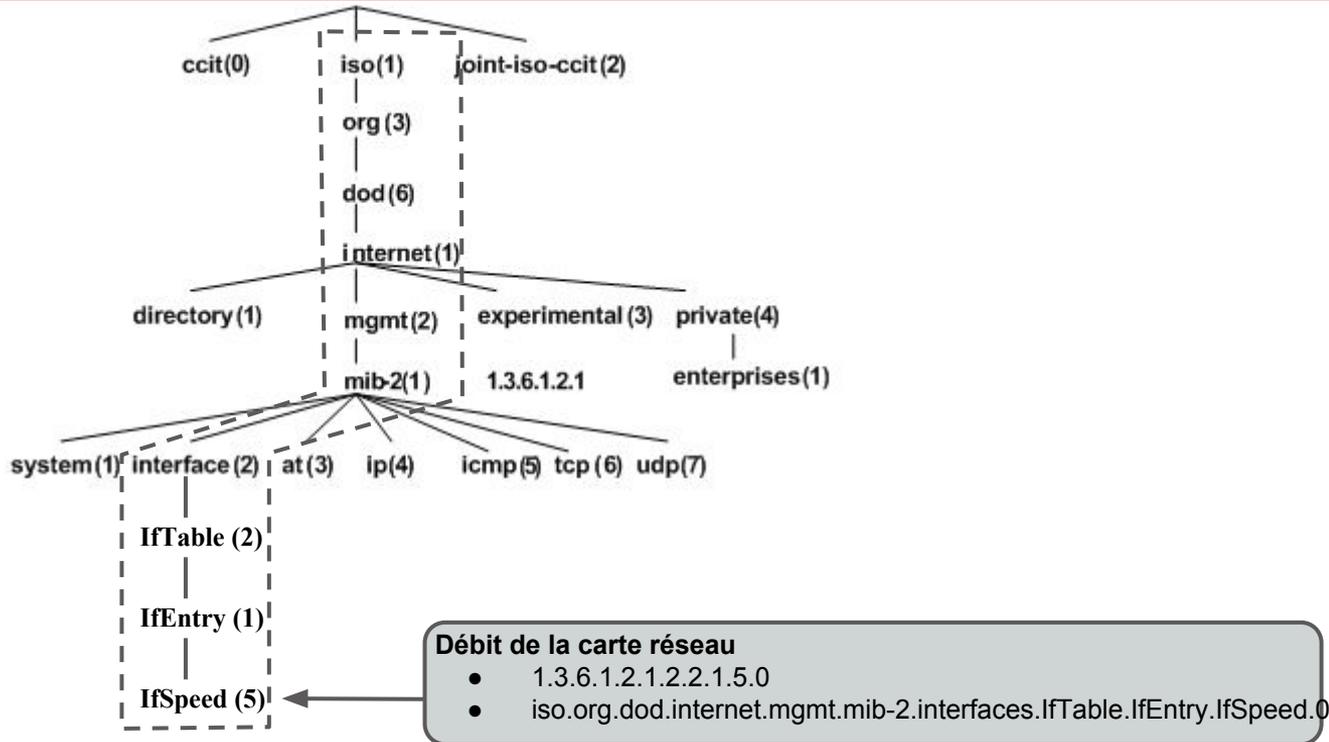
- liste des noeuds de la MIB traversés pour atteindre la variable
  - Un OID unique pour chaque variable (structure en arbre)
  - Sous arbre de la MIB
    - ensemble de variables qui traitent du même domaine
    - ex : sous-arbre ip, system, interfaces, ...
-

# 3 - Supervision basée sur SNMP

---

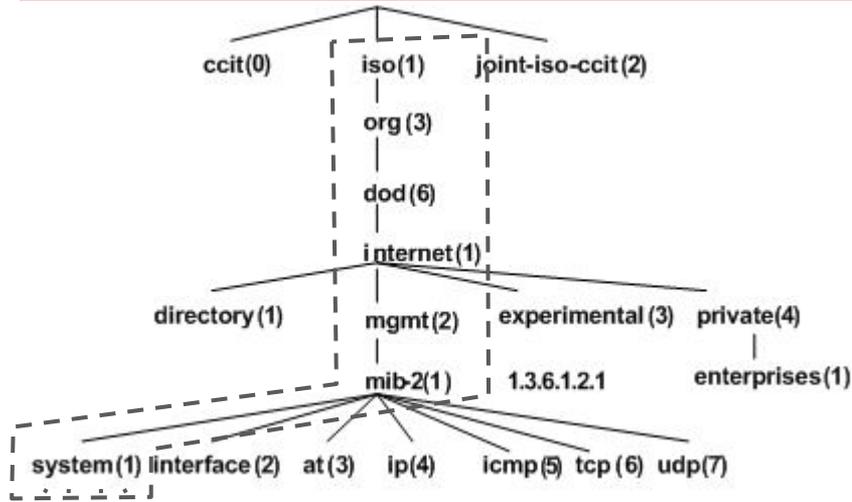
- Exemples d'OID :
    - variable “débit carte réseau”
      - 1.3.6.1.2.1.2.2.1.5.0 en notation numérique
      - iso.org.dod.internet.mgmt.mib-2.interfaces.IfTable.IfEntry.IfSpeed.0 en notation symbolique
    - sous-arbre “system”
      - 1.3.6.1.2.1.1 en notation numérique
      - iso.org.dod.internet.mgmt.mib-2.system en notation symbolique
-

# 3 - Supervision basée sur SNMP



# 3 - Supervision basée sur SNMP

---



## Sous-arbre "system"

- 1.3.6.1.2.1.1
- iso.org.dod.internet.mgmt.mib-2.interfaces.IfTable.IfEntry.IfSpeed.

# 3 - Supervision basée sur SNMP

---

## 3.3.3 - Différentes instances d'une variable

- Une même variable peut avoir plusieurs instances
  - ex : une valeur de IfSpeed par carte réseau !
- Si une variable n'a qu'UNE seule instance, on rajoute 0 à la fin de l'OID
  - ex : 1.3.6.1.2.1.2.2.1.5.0 pour la variable IfSpeed si

# 3 - Supervision basée sur SNMP

---

- Si une variable a plusieurs instances, pour accéder à l'une de ces instances, on rajoute son numéro à la fin de l'OID
    - ex : 1.3.6.1.2.1.2.2.1.5.**1** pour la carte réseau n°1
    - ex : 1.3.6.1.2.1.2.2.1.5.**2** pour la carte réseau n°2
    - ...
  - Le numéro affecté à chaque carte réseau est donné dans 1.3.6.1.2.1.2.2.1.1 (IfIndex)
-

# 3 - Supervision basée sur SNMP

---

## 3.3.4 - Différents types de variables

- La description des variables (syntaxe) de la MIB est normalisée
    - ASN-1
  - Plusieurs types
    - INTEGER = entiers
    - STRING = chaîne de caractère
    - SEQUENCE = structure contenant plusieurs variables
-

# 3 - Supervision basée sur SNMP

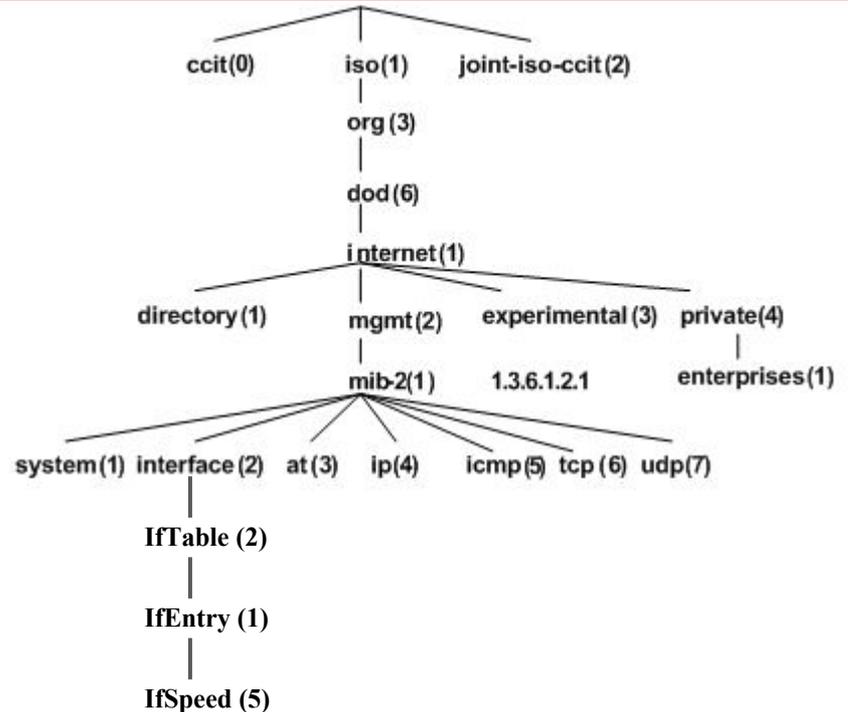
---

- SEQUENCE OF = tableau de variables de même type
  - IP\_ADDRESS = adresse IP
  - TIMESTICKS = compteur de temps
  - COUNTER = compteur croissant entier
  - GAUGE = compteur qui peut croître et décroître
-

# 3 - Supervision basée sur SNMP

## Exemples :

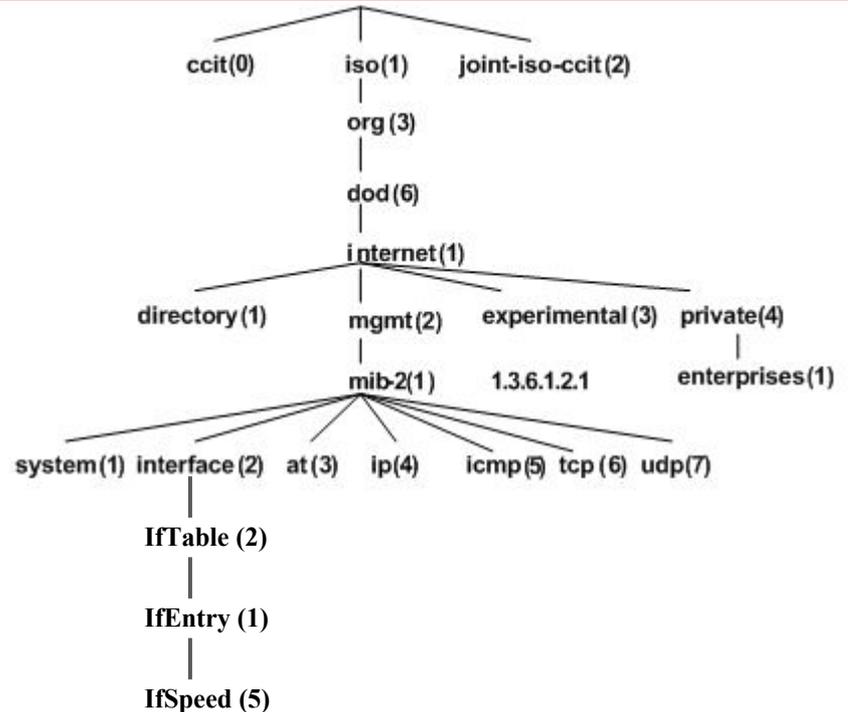
mib-2	OBJECT IDENTIFIER ::= { mgmt 1 }
system	OBJECT IDENTIFIER ::= { mib-2 1 }
interfaces	OBJECT IDENTIFIER ::= { mib-2 2 }
at	OBJECT IDENTIFIER ::= { mib-2 3 }
ip	OBJECT IDENTIFIER ::= { mib-2 4 }
icmp	OBJECT IDENTIFIER ::= { mib-2 5 }
tcp	OBJECT IDENTIFIER ::= { mib-2 6 }
udp	OBJECT IDENTIFIER ::= { mib-2 7 }
egp	OBJECT IDENTIFIER ::= { mib-2 8 }
transmission	OBJECT IDENTIFIER ::= { mib-2 10 }
snmp	OBJECT IDENTIFIER ::= { mib-2 11 }



# 3 - Supervision basée sur SNMP

## Exemples :

```
IfEntry ::=  
  SEQUENCE {  
    ifIndex      INTEGER,  
    ifDescr     DisplayString,  
    ifType      INTEGER,  
    ifMtu       INTEGER,  
    ifSpeed     Gauge,  
    ifPhysAddress PhysAddress,  
    ifAdminStatus INTEGER,  
    ifOperStatus INTEGER,  
    ifLastChange TimeTicks,  
  }
```

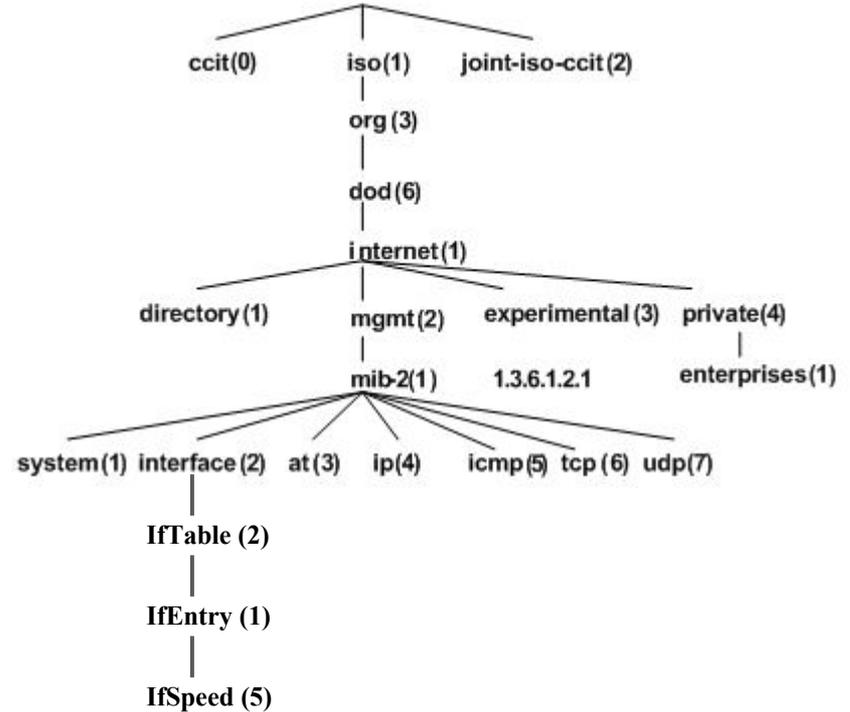


# 3 - Supervision basée sur SNMP

## Exemples :

ifInOctets	Counter,
ifInUcastPkts	Counter,
ifInNUcastPkts	Counter,
ifInDiscards	Counter,
ifInErrors	Counter,
ifInUnknownProtos	Counter,
ifOutOctets	Counter,
ifOutUcastPkts	Counter,
ifOutNUcastPkts	Counter,
ifOutDiscards	Counter,
ifOutErrors	Counter,
ifOutQLen	Gauge,
ifSpecific	OBJECT IDENTIFIER

}



# 3 - Supervision basée sur SNMP

---

## 3.3.5 - MIB normalisée vs MIB privée

- MIB normalisée
    - Tout ce qui est dans le sous-arbre 1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2)
    - liste des variables et descriptif normalisé, donc identique sur tous les matériels
    - Rq : pas toujours implémenté et respecté :-)
-

# 3 - Supervision basée sur SNMP

---

- MIB privée
    - Une partie de la MIB est laissée à la discrétion des constructeurs
      - permet d'avoir une MIB spécifique à chaque matériel
    - le sous-arbre privé est à l'OID 1.3.6.1.4.1 (iso.org.dod.internet.private.entreprises)
    - Ex: quantité de RAM utilisée, nbre de processus actifs, ...
-

# 3 - Supervision basée sur SNMP

---

- Nécessité d'avoir un descriptif de cette partie pour chaque machine
  - normalement mis à disposition par le constructeur (sinon ça ne sert à rien !!!)
  - on trouve ces fichiers au format ASN-1 sur les sites des constructeurs
  - A installer sur le manager pour qu'il puisse déchiffrer cette partie de la MIB
  - Rq : ne pas avoir ces fichiers n'interdit pas de se

# 3 - Supervision basée sur SNMP

---

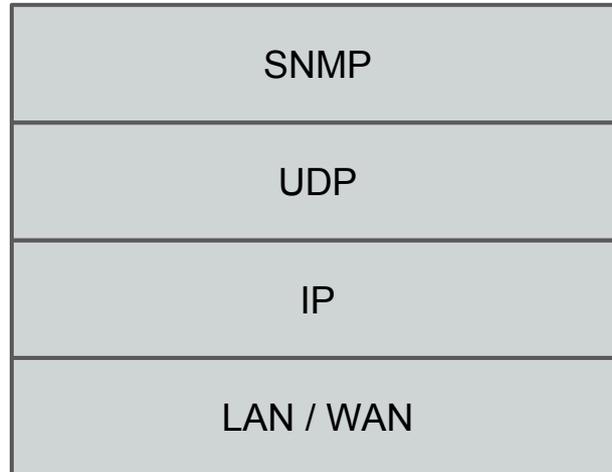
- Résumé
    - on trouve les OID “classiques” dans la partie normalisée de la MIB
      - 1.3.6.1.2.1.2
    - Si on veut une variable plus précise, il faut fouiller dans la partie privée (à condition que cette variable existe !)
      - 1.3.6.1.4.1
-

# 3 - Supervision basée sur SNMP

---

## 3.4 INTERACTIONS MANAGER - AGENT

- Pile de protocoles pour la communication



# 3 - Supervision basée sur SNMP

---

- 3 grands groupes d'interactions
    - **requêtes** du manager vers l'agent
    - **réponses** d'un agent au manager suite à une requête
    - **notifications** d'un agent à un manager, sans requête préalable
-

# 3 - Supervision basée sur SNMP

---

- Requêtes d'un manager à un agent
    - requêtes envoyées sur port UDP 161 de l'agent
    - Différents types de requêtes
      - **GET (get-request)** :
        - demande de lecture d'une variable (l'OID de la variable précisé)
      - **GET-NEXT (get-next-request)** :
        - demande de lecture de toutes les variables d'un sous-arbre (OID du sous-arbre précisé)
      - **SET (set-request)** :
        - demande d'écriture dans une variable (OID précisé)
-

# 3 - Supervision basée sur SNMP

---

- Réponses d'un agent à un manager, suite à une requête
    - envoi sur le port 162 du manager
    - un seul type de réponse
      - **RESPONSE (get-response) :**
        - réponse à la requête précédente (OID + valeur précisés)
    - Si requête GET-NEXT, l'agent peut faire une réponse différente par variable, ou une réponse unique contenant toutes les variables et leurs valeurs
-

# 3 - Supervision basée sur SNMP

---

- Notification d'un agent à un manager, sans requête préalable
    - notification = alerte de dépassement de seuil pour une variable (TRAP)
    - Les TRAPs doivent être configurés par l'admin. sur le matériel (pas de TRAP actif par défaut)
    - envoi sur port 162 du manager (comme réponses aux requêtes)
-

# 3 - Supervision basée sur SNMP

---

- Notification d'un agent à un manager, sans requête préalable
    - 7 différents types de TRAPs
      - Cold Start / Warm Start
      - Link Down, Link Up
      - Authentication Failure
      - EGP Neighbor Loss
      - **Enterprise Specific**
-

# 3 - Supervision basée sur SNMP

---

- Notification d'un agent à un manager, sans requête préalable
    - très utile pour avoir une notification de problème rapidement
    - Ex :
      - espace disque disponible  $<$  seuil sur un serveur
      - nbre paquets suppr (drop)  $>$  seuil sur routeur
      - nbre processus  $>$  seuil sur serveur
      - débit carte réseau  $>$  seuil sur routeur
-

# 3 - Supervision basée sur SNMP

---

## 3.5 CONTROLE D'ACCES DANS SNMP

- 3 version différentes de SNMP
    - v1, v2c, v3
    - la plus propagée aujourd'hui = v1
  - Différences majeures entre les versions
    - Contrôle d'accès différent entre v1/v2c et v3
    - MIB plus développée en v2c/v3 qu'en v1
-

# 3 - Supervision basée sur SNMP

---

## 3.5.1 - Contrôle d'accès en v1 et v2c

- Basé sur la notion de communauté
  - Communauté = ensemble de droits d'accès à la MIB
    - type d'accès (RO / RW)
    - parties de la MIB "visibles"
    - machines autorisées à utiliser cette communauté (sources autorisées)
-

# 3 - Supervision basée sur SNMP

---

- Il suffit de donner le nom de la communauté que l'on veut utiliser dans les requêtes SNMP pour obtenir les droits associés
    - pas de mot de passe !
    - la sécurité ne repose que sur la connaissance ou non du nom de la communauté
-

# 3 - Supervision basée sur SNMP

---

- Exemples
    - communauté PUBLIC
      - type d'accès = RO
      - parties de la MIB visibles = toute la MIB
      - Machines autorisées = all
  
    - exemple de commande utilisant la communauté  
`snmpwalk -c public -v 2c 10.2.18.1 1.3.6.1.2`
-

# 3 - Supervision basée sur SNMP

---

- Exemples

- communauté PRIVATE

- type d'accès = RW

- parties de la MIB visibles = 1.3.6.1.2

- Machines autorisées = all

- exemple de commande utilisant la communauté

- ```
snmpwalk -c private -v 2c 10.2.18.1 1.3.6.1.2
```

---

# 3 - Supervision basée sur SNMP

---

- Par défaut :
    - communautés PUBLIC et PRIVATE existent sur les matériels ayant un agent SNMP
      - n'importe qui peut y accéder !
    - C'est à l'admin. réseau de restreindre leur accès s'il le souhaite
    - De plus en plus souvent, l'accès est interdit à toutes les machines par défaut dans ces deux communauté
-

# 3 - Supervision basée sur SNMP

---

## 3.5.2 - Contrôle d'accès en v3

- **Totalement différent**
    - plus de notion de communauté
  - **Basé sur une authentification classique**
    - login + password
    - droits d'accès : RO/RW + visibilité dans la MIB
    - machines sources autorisées
-

# 3 - Supervision basée sur SNMP

---

- Les droits d'accès sont donnés à chaque utilisateur (à chaque login)
  - Cryptage possible des communications manager - agent
    - confidentialité des données échangées
  - Sécurité beaucoup avancée qu'en v1/v2c
    - mais peu utilisé car peu déployé sur les matériels (notamment les matériels anciens)
-

# 3 - Supervision basée sur SNMP

---

## 3.6 COMMENT RECUPERER LA VALEUR DE VARIABLES D'UNE MIB ?

- On peut exécuter “à la main” des commandes pour “fouiller” dans la MIB d'un matériel
- Pour des raisons de temps, nous ne détaillons que les commandes dans un environnement Linux (Windows très voisin)

# 3 - Supervision basée sur SNMP

---

- Obtention de la valeur d'une variable

```
snmpget -Ou -v 2c -c public 10.2.18.1 1.3.6.1.2.1.1.1.0
```

avec -Ou : affichage du résultat en référence symbolique (-On pour affichage en numérique)

-v 2c : snmp version 2c

-c public : utilisation des droits de la communauté "public"

10.2.18.1 : @IP de l'agent questionné (routeur IUT)

1.3.6.1.2.1.1.1.0 : OID recherché (sysDescr)

---

# 3 - Supervision basée sur SNMP

---

```
root@gallon-laptop:/home/gallon# snmpget -Ou -c public -v 2c 127.0.0.1 1.3.6.1.2.1.1.1.0
system.sysDescr.0 = STRING: Linux gallon-laptop 2.6.32-24-generic #43-Ubuntu SMP Thu Sep 16 14:17:33 UTC 2010 i686
root@gallon-laptop:/home/gallon# █
```

```
root@gallon-laptop:/home/gallon# snmpget -Ou -c public -v 2c 127.0.0.1 1.3.6.1.2.1.1.3.0
system.sysUpTime.sysUpTimeInstance = Timeticks: (11004) 0:01:50.04
root@gallon-laptop:/home/gallon# █
```

---

# 3 - Supervision basée sur SNMP

---

- Exploration d'un sous-arbre

```
snmpwalk -Ou -v 2c -c public 10.2.18.1 1.3.6.1.2.1.1.1.0
```

avec -Ou : affichage du résultat en référence symbolique (-On pour affichage en numérique)

-v 2c : snmp version 2c

-c public : utilisation des droits de la communauté "public"

10.2.18.1 : @IP de l'agent questionné (routeur IUT)

1.3.6.1.2.1.1.1.0 : OID recherché (sysDescr)

---

# 3 - Supervision basée sur SNMP

---

```
root@gallon-laptop:/home/gallon# snmpwalk -Ou -c public -v 2c 127.0.0.1 1.3.6.1.2.1
system.sysDescr.0 = STRING: Linux gallon-laptop 2.6.32-24-generic #43-Ubuntu SMP Thu Sep 16 14:17:33 UTC 2010 i686
system.sysObjectID.0 = OID: enterprises.netSnmp.netSnmpEnumerations.netSnmpAgentOIDs.10
system.sysUpTime.sysUpTimeInstance = Timeticks: (33593) 0:05:35.93
system.sysContact.0 = STRING: Root <root@localhost> (configure /etc/snmp/snmpd.local.conf)
system.sysName.0 = STRING: gallon-laptop
system.sysLocation.0 = STRING: Unknown (configure /etc/snmp/snmpd.local.conf)
system.sysORLastChange.0 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORID.1 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpFrameworkMIB.snmpFrameworkMIBConformance.snmpFrameworkMIBCompliances.snmpFrameworkMIBCompliance
system.sysORTable.sysOREntry.sysORID.2 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpMPDMIB.snmpMPDMIBConformance.snmpMPDMIBCompliances.snmpMPDMIBCompliance
system.sysORTable.sysOREntry.sysORID.3 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpUsmMIB.usmMIBConformance.usmMIBCompliances.usmMIBCompliance
system.sysORTable.sysOREntry.sysORID.4 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB
system.sysORTable.sysOREntry.sysORID.5 = OID: tcpMIB
system.sysORTable.sysOREntry.sysORID.6 = OID: ip
system.sysORTable.sysOREntry.sysORID.7 = OID: udpMIB
system.sysORTable.sysOREntry.sysORID.8 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpVacmMIB.vacmMIBConformance.vacmMIBGroups.vacmBasicGroup
system.sysORTable.sysOREntry.sysORDescr.1 = STRING: The SNMP Management Architecture MIB.
system.sysORTable.sysOREntry.sysORDescr.2 = STRING: The MIB for Message Processing and Dispatching.
system.sysORTable.sysOREntry.sysORDescr.3 = STRING: The management information definitions for the SNMP User-based Security Model.
system.sysORTable.sysOREntry.sysORDescr.4 = STRING: The MIB module for SNMPv2 entities
system.sysORTable.sysOREntry.sysORDescr.5 = STRING: The MIB module for managing TCP implementations
system.sysORTable.sysOREntry.sysORDescr.6 = STRING: The MIB module for managing IP and ICMP implementations
system.sysORTable.sysOREntry.sysORDescr.7 = STRING: The MIB module for managing UDP implementations
system.sysORTable.sysOREntry.sysORDescr.8 = STRING: View-based Access Control Model for SNMP.
system.sysORTable.sysOREntry.sysORUpTime.1 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.2 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.3 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.4 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.5 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.6 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.7 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.8 = Timeticks: (3) 0:00:00.03
system.sysORTable.sysOREntry.sysORUpTime.8 = No more variables left in this MIB View (It is past the end of the MIB tree)
root@gallon-laptop:/home/gallon#
```

# 3 - Supervision basée sur SNMP

---

- Traduction OID numérique - OID symbolique

snmptranslate -Ou 1.3.6.1.2.1.1.1.0

snmptranslate -On iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0

avec -Ou/-On : affichage du résultat en référence symbolique /  
numérique

---

# 3 - Supervision basée sur SNMP

---

```
root@gallon-laptop:/home/gallon# snmptranslate -Ou 1.3.6.1.2.1.1.1.0  
system.sysDescr.0
```

```
root@gallon-laptop:/home/gallon# snmptranslate -Ouf 1.3.6.1.2.1.1.1.0  
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0
```

```
root@gallon-laptop:/home/gallon# snmptranslate -On iso.org.dod.internet.mgmt.mib-2.system  
.1.3.6.1.2.1.1
```

# 3 - Supervision basée sur SNMP

---

- Affichage de la structure d'un sous-arbre

```
snmptranslate -Tp 1.3.6.1.2.1.1
```

```
snmptranslate -Tp iso.org.dod.internet.mgmt.mib-2.system
```

---

# 3 - Supervision basée sur SNMP

```
root@gallon-laptop:/home/gallon# snmptranslate -Tp 1.3.6.1.2.1.1
+--system(1)
|
+-- -R-- String    sysDescr(1)
|   Textual Convention: DisplayString
|   Size: 0..255
+-- -R-- ObjID    sysObjectID(2)
+-- -R-- TimeTicks sysUpTime(3)
|
|   +--sysUpTimeInstance(0)
|
+-- -RW- String    sysContact(4)
|   Textual Convention: DisplayString
|   Size: 0..255
+-- -RW- String    sysName(5)
|   Textual Convention: DisplayString
|   Size: 0..255
+-- -RW- String    sysLocation(6)
|   Textual Convention: DisplayString
|   Size: 0..255
+-- -R-- INTEGER   sysServices(7)
|   Range: 0..127
+-- -R-- TimeTicks sysORLastChange(8)
|   Textual Convention: TimeStamp
|
+--sysORTable(9)
|
+--sysOREntry(1)
|   Index: sysORIndex
|
+-- ---- INTEGER   sysORIndex(1)
|   Range: 1..2147483647
+-- -R-- ObjID    sysORID(2)
+-- -R-- String    sysORDescr(3)
|   Textual Convention: DisplayString
|   Size: 0..255
+-- -R-- TimeTicks sysORUpTime(4)
|   Textual Convention: TimeStamp
root@gallon-laptop:/home/gallon#
```

# 3 - Supervision basée sur SNMP

---

## 3.7 SUIVRE L'EVOLUTION D'UNE VARIABLE

- Parfois, récupérer la valeur d'une variable n'est pas très parlant
  - C'est son évolution dans le temps qui est significatif
    - ex : fluctuation du débit de la carte réseau plus intéressant que sa valeur instantanée
-

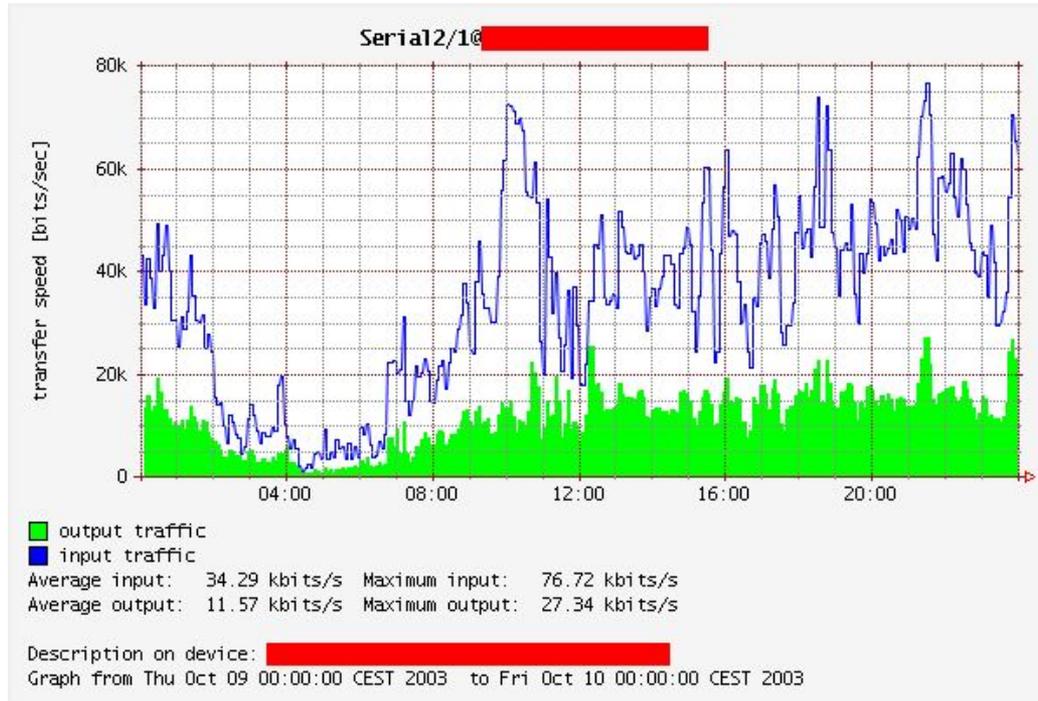
# 3 - Supervision basée sur SNMP

---

- Il existe des logiciels qui tracent des courbes de variation des variables
    - récupèrent régulièrement (polling) la valeur de la variable
    - trace la courbe au fur et à mesure
  - Exemples de logiciels
    - MRTG
    - RDDTOOLS
-

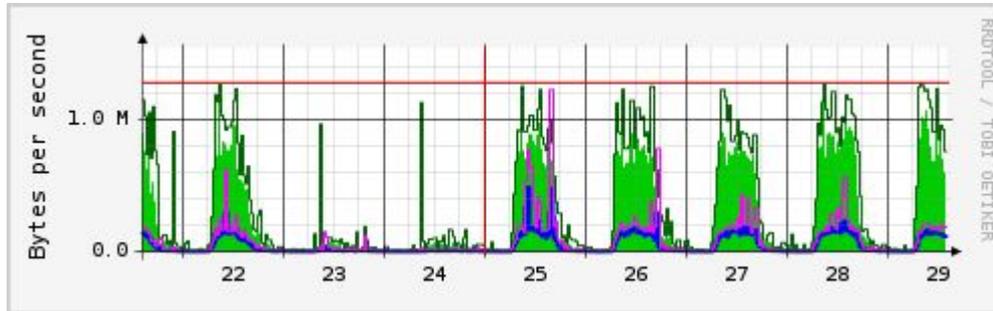
# 3 - Supervision basée sur SNMP

---

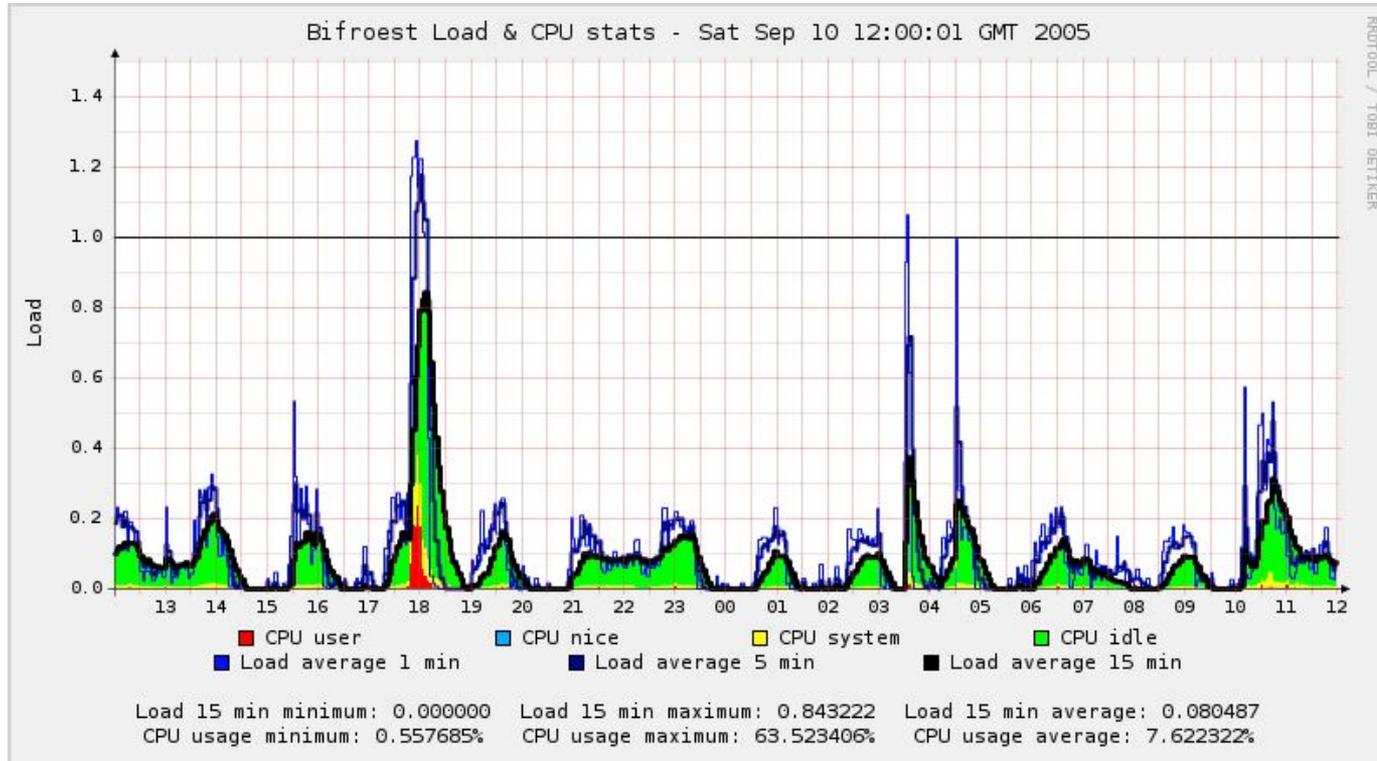


# 3 - Supervision basée sur SNMP

---



# 3 - Supervision basée sur SNMP



# PLAN

---

- 1 - Introduction
  - 2 - Cartographie de présence
  - 3 - Supervision basée sur SNMP
  - 4 - Supervision basée sur SYSLOG**
  - 5 - Quelques logiciels de supervision
-

# 4 - supervision basée sur SYSLOG

---

## 4.1 GENERALITES

- Toutes les machines maintiennent des fichiers de log de leurs principales opérations = fichiers textes dans lesquels apparaissent les événements importants
-

# 4 - supervision basée sur SYSLOG

---

- Pourquoi des fichiers de log ?
    - retrouver des traces de problèmes lors de l'installation / la configuration / le fonctionnement d'un matériel ou application
    - trace des opérations des utilisateurs
    - trace des tentatives d'intrusion
    - audit des matériel
    - ...
-

# 4 - supervision basée sur SYSLOG

---

- Problèmes

- beaucoup d'applications => beaucoup de fichiers de logs
  - la taille des fichiers de logs peut être très grande
  - les infos utiles sont noyées dans une somme d'informations peu importantes
-

# 4 - supervision basée sur SYSLOG

---

- Une solution : Syslog
    - centralise les fichiers de logs des différents matériels et applications : PCs, serveurs, matériels actifs, ...
    - filtrage des messages pour ne conserver que ce qui est important
  - Par défaut
    - Syslog peut être trouvé sur quasiment toutes les machines réseau
-

# 4 - supervision basée sur SYSLOG

---

## 4.2 STRUCTURE DES MESSAGES

- Malheureusement variable en fonction de l'implémentation
  - Champs fréquemment rencontrés :
    - facility
      - identifiant (0..23) du composant / application ayant généré le message
-

# 4 - supervision basée sur SYSLOG

---

- ex : 0 → kernel  
2 → mail  
16..23 (local0 .. local7) → réservé à la configuration locale (application particulières aux entreprises)
-

# 4 - supervision basée sur SYSLOG

---

- security level
  - 0 = emerg → alerte maximale (le syst va s'arrêter)
  - 1 = alert → nécessite une action immédiate (espace disque full, ...)
  - 2 → crit → fonctionnement incorrect du syst / de l'application (disk failure, ...)
  - 3 = err → problème grave, mais qui ne remet pas en cause le fonctionnement du syst / appli (conflit

# 4 - supervision basée sur SYSLOG

---

- security level
    - 4 → warning → messages d'alertes (bail DHCP terminé, ...)
    - 5 → notice → message d'alertes (différence avec warning pas claire)
    - 6 → information (l'application a démarrée, ...)
    - 7 → debug → message de débogage d'applications
-

# 4 - supervision basée sur SYSLOG

---

- Classification des messages de syslog
    - basé sur facility.severity
      - ex : kernel.warning est plus prioritaire que mail.crit
      - on peut aussi utiliser la wildcard \* . ex : \*.err représente tous les messages de niveau err quelle que soit leur facility
  - Timestamp
    - horodatage du message
    - Ce champs est très utile, mais peut être faux si l'heure et la date sur le matériel sont faux
      - utilisation de NTP pour synchroniser l'heure des matériels
-

# 4 - supervision basée sur SYSLOG

---

- Host
    - nom ou @IP de l'émetteur du message
  - Tag
    - nom du processus ayant généré le message
  - Message
    - texte ascii donnant des éléments sur l'événement qui s'est produit
    - normalement compréhensible, mais ...
-

# 4 - supervision basée sur SYSLOG

---

- Résumé

< facility, security, timestamp, host, tag,   
texte >

---

# 4 - supervision basée sur SYSLOG

---

- Remarque importante
    - 1 message syslog ne peut pas dépasser 1kb !
    - on ne peut mettre que très peu d'informations dans chaque message
    - La plupart du temps, le message ne contient que le triplet `facility.severity.message`
-

# 4 - supervision basée sur SYSLOG

---

## 4.3 FONCTIONNEMENT DE SYSLOG

- Syslog server (daemon)
    - processus Syslog qui récupère les messages en provenance des différents clients Syslog, pour les traiter (filtrage, ...)
    - A priori, il ne génère pas de nouveau message. Il ne fait que traiter (avant d'envoyer éventuellement à un autre serveur) les messages Syslog qu'il reçoit
-

# 4 - supervision basée sur SYSLOG

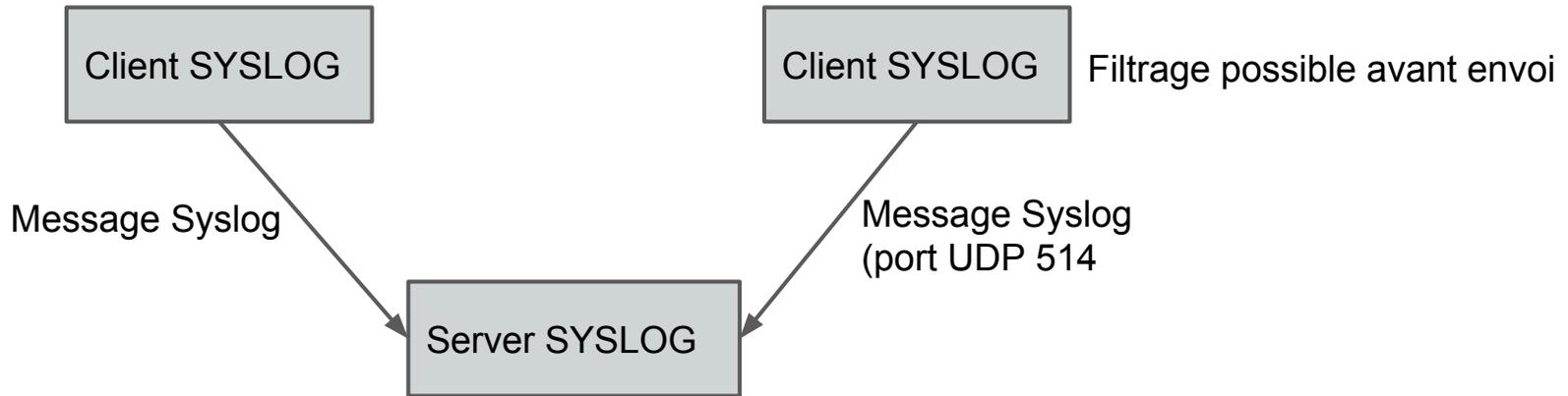
---

- Syslog client
    - processus Syslog sur une machine qui génère les messages Syslog de la machine
    - N'envoie pas forcément tous ses événements sur le serveur Syslog. Il peut faire un filtrage à la source
-

# 4 - supervision basée sur SYSLOG

---

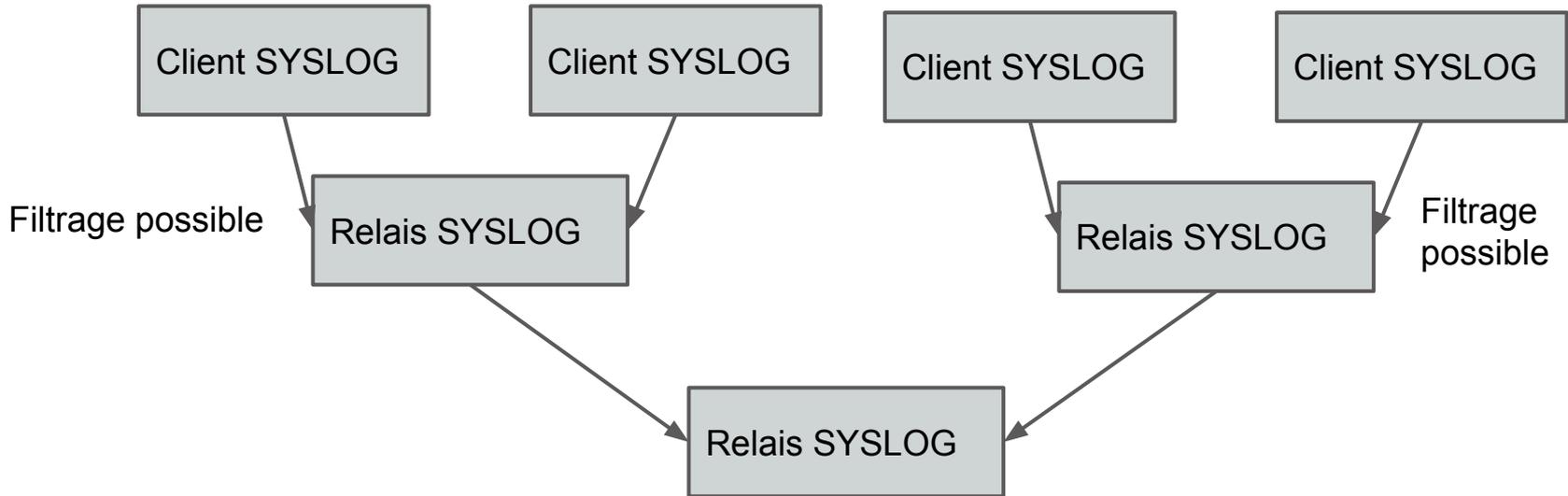
- Exemple d'architecture



# 4 - supervision basée sur SYSLOG

---

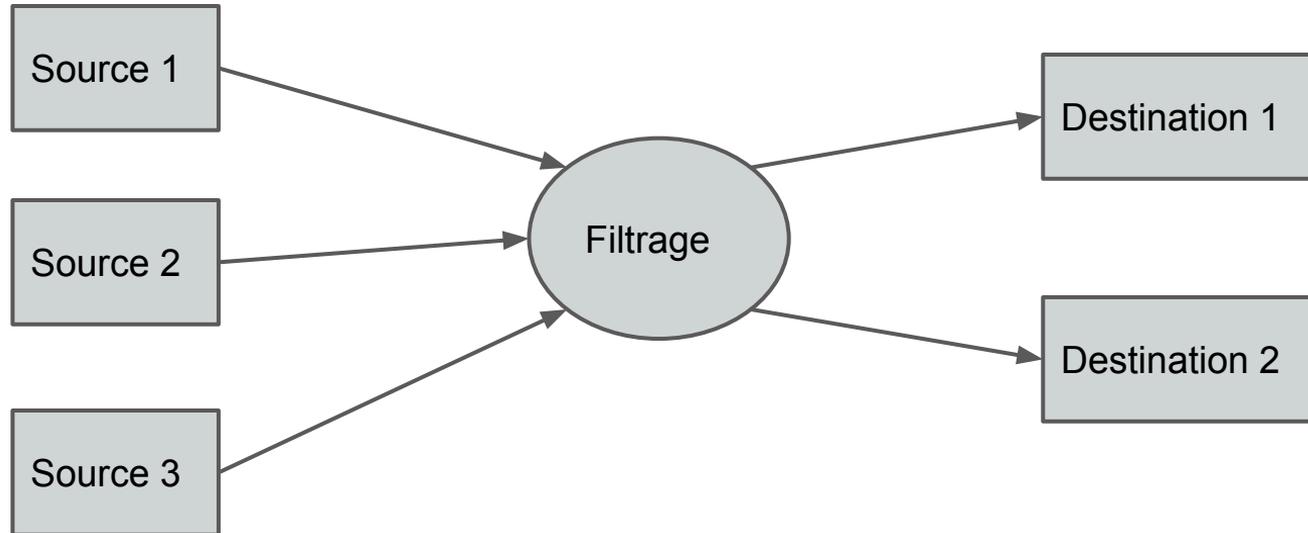
- Exemple d'architecture



# 4 - supervision basée sur SYSLOG

---

- Fonctionnement d'un client / serveur



# 4 - supervision basée sur SYSLOG

---

- Fonctionnement d'un client / serveur
    - Sources
      - désignent les provenance des messages à filtrer
        - fichiers de logs sur la machine
        - flux réseau en provenance d'autres matériels
    - Filtrage
      - règles permettant de dire si on conserve ou pas les messages des sources
        - Configuré par l'administrateur
        - demande beaucoup d'expérience pour savoir quoi garder
-

# 4 - supervision basée sur SYSLOG

---

- Fonctionnement d'un client / serveur
  - Destinations
    - “endroits” de stockage des messages conservés après filtrage
      - fichier de log
      - machine distante (serveur syslog par exemple)
      - un mail
      - un sms
      - affichage à l'écran
      - ...

# PLAN

---

1 - Introduction

2 - Cartographie de présence

3 - Supervision basée sur SNMP

4 - Supervision basée sur SYSLOG

**5 - Quelques logiciels de supervision**

---

# 5 - Quelques logiciels de supervision

---

- Logiciels snmp de base
    - snmp
    - snmpd (agent)
    - net-snmp (windows)
  - Logiciels Syslog
    - syslog-ng
    -
-

# 5 - Quelques logiciels de supervision

---

- Logiciels Browser de Mib
  - mbrowse
  - snmpb
  - Mib-Browser (Windows)

# 5 - Quelques logiciels de supervision

---

- Logiciels de supervision
    - Nagios
    - Shinken
    - Centreon
    - HP Openview
-