Université de Savoie Mont Blanc
Master 1 Télécoms Réseaux Informatiques

2023-2024

ETRS 813_TRI
Supervision

Cours : 7 x 1.5H
TD : 4 x 1,5H
TP : 14 H

Florent Lorne
florent.lorne@univ-smb.fr
Bureau : Chablais 13

Introduction

- Entreprises dépendantes de leur système d'information.
  - perte ou ralentissement d'une application/service
  - ==> grandes conséquences sur l'activité de l'entreprise

- Très important de connaître :
  - la disponibilité et la qualité d'un service rendu
  - ==> pouvoir réagir immédiatement en cas de panne

- Protocoles / logiciels / outils de supervision
  - Permettre aux administrateurs d'avoir une idée en temps réel de l'état de l'ensemble des machines et des services.
  - En cas de problèmes, l'administrateur peut recevoir SMS, e-mail, messages instantanés…,
  - ==> réduire très significativement le délai d'indisponibilité

- Développement de l'infogérance -> confier à un prestataire la gestion de tout ou partie du système d'information (SI) d'une entreprise.
  - Définition pour chaque service de SLA (Service Level Agreement – Accords de niveau de service) :
    - Disponibilité (availability)
    - Response time, solution time ...
  - Infogérance nécessite :
    - **Supervision**
    - **Reporting** : création de rapports pour les clients synthétisant les métriques de bonnes gestions du SI (availability, response time ...).

Introduction

Aspects normalisation

- La supervision peut être considérée comme une « sous-branche » de l'administration réseau et système.

- (1990) l'ISO décrit des fonctions administratives liées à la gestion des SI, regroupées en cinq familles :
  - la gestion des configurations matérielles : Configuration Management
  - les pannes et incidents : Fault Management
  - les performances : Performance Management
  - la sécurité : Security Management
  - Le taux d'utilisation : Accounting Management

- Pour chacune de ces activités on distingue deux approches complémentaires :
  - Approche technique (configuration, rédaction des documentations et procédures ...)
  - Approche administrative/managériale (achat, évaluation des coûts, formations, inventaires …)

- Supervision, à l'instar de la sécurité : activité transversale
  - Rôle principal de la supervision : gérer les pannes/ incidents et les performances.
  - Mais elle intervient aussi dans :
    - la **sécurité** ex : la détection d'intrusion peut être considérée comme de la supervision,
    - la **gestion** des configurations matérielles/logicielles. ex : une base de données recense serveurs, applications, matériels réseaux … CMDB (Configuration management database)
    - Aspects **financiers** : évaluer le coût d'usage d'une ressource (service de fichiers, impression …). 2 paramètres essentiels : temps d'utilisation et volume d'informations.

Introduction

- Autre approche pour définir des bonnes méthodes de gestion des services informatiques : le référentiel **ITIL** (**I**nformation **T**echnology **I**nfrastructure **L**ibrary).
  Ce référentiel de gestion des processus informatiques a pour pilier le « service delivery » qui inclut entre autres les concepts de :
  - Service Level Management notion de SLA Service Level Agreement
  - Availability Management
  - Capacity Management

- Aspects plus éloignés de la supervision mais liés :
  - Reprise d'activité en cas de pannes :
    - Que veut-on récupérer et en combien de temps ?

RTO – Recovery Time Objective

RPO – Recovery Point Objective

  - Haute disponibilité
    - Définir la chaîne complète pour assurer un service rendu
    - Identifier les points durs : SPOF

Introduction

Définition

- Superviser :
  - Définition générale : contrôler ou surveiller sur des points essentiels un travail effectué (superviser un projet)
  - Supervision réseau/service : contrôler en temps réel la qualité et l'état des réseaux/hôtes/services.

- Objectifs opérationnels de la supervision :
  - Être informé du bon fonctionnement des éléments actifs du SI (Monitoring).
  - Instrumenter et mesurer les capacités / le comportement des services et réseaux.
  - Analyser les mesures collectées, repérer (si possible avant les usagers) des comportements habituels (normaux) / inhabituels (anormaux).
  - Gérer de manière optimale (automatique si possible) le traitement des pannes/incidents et la qualité des services
  - Aider aux diagnostics des pannes (trouver rapidement la cause d'un problème)
  - Éviter l'effet domino (une avarie en entraînant d'autres).
  - Optimiser les performances
  - Prévoir, avoir une gestion proactive, découvrir les signes annonciateurs. :
    - planifier les futures évolutions nécessaires,
    - anticiper les pannes qui peuvent se produire.
  - Mesurer les effets de l'installation d'un nouveau logiciel ou d'un nouveau matériel.
  - Justifier auprès des décideurs l'utilisation de l'argent investi dans le SI et l'infrastructure réseau.
  - Dimensionner correctement le système d'informations
  - Ne pas avoir uniquement le ressenti des utilisateurs. Utilisateur : moyen de supervision peu fiable et pas toujours agréable

Introduction

Exemples :

Disponibilité : Un serveur d'authentification tombe en panne un dimanche après midi. Si pas de supervision --> l'administrateur n'est pas prévenu. Pb lorsque les usagers arrivent le lundi matin avant lui.

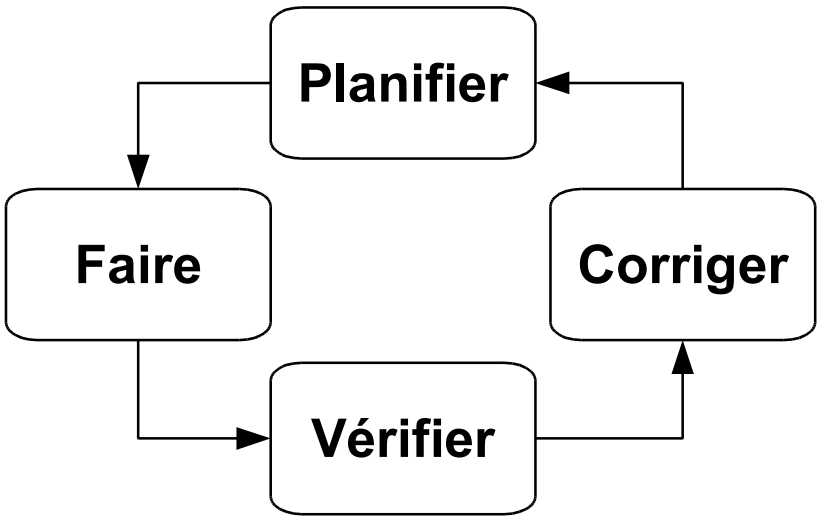
Cas d'un disque où s'effectuent des sauvegardes, et qui serait plein. Cette sauvegarde défaillante peut empêcher une application voire un serveur de fonctionner.
L'administrateur constate qu'un serveur est non disponible. Il regarde dans l'outil de supervision depuis quand il est dans cet état. Il s'aperçoit alors que peu de temps avant il y avait une alerte de disque plein.

Installation d'un matériel de compression de flux entre 2 sites. Vous voulez connaître ses performances. Nécessité d'avoir fait des mesures avant installation et après. Ne surtout pas se baser sur des impressions (subjectives). Avoir des données exprimées avec des métriques normalisées.

Introduction

- Plan du cours :
  - Première partie : méthodologie à appliquer pour mettre en œuvre une solution de supervision du système d'informations.
    - Présentation rapide des différentes méthodes et outils disponibles pour faire de la supervision
  - Deuxième partie : présentation plus complète des outils et protocoles de supervision :
    - protocole SNMP
    - logiciel de supervision Nagios, Zabbix.
  - Troisième partie : sous-partie de la supervision : la métrologie réseau.
    - métriques
    - outils de mesures
    - résolution des problèmes de performance de bout en bout
  - Quatrième partie :
    - Haute disponibilité
    - Législation et traitement des informations liées aux utilisateurs.

Méthodologie



Planification – To plan (travail de bac +5)

- Définir
  - ce qui doit être supervisé
  - comment le superviser
  - ce qui doit être fait en cas de problème
  - combien ça va coûter, combien ça va rapporter (ROI – Return Of Invest)

- Grandes étapes lors de la planification (à compléter)
  -

Méthodologie

- Nécessite d'avoir une bonne expérience du fonctionnement du SI pour remonter les informations les plus pertinentes.

- Attention à la tentation de tout superviser. Si un indicateur n'apporte aucune information par rapport à un autre déjà en place, il ne faut tout simplement pas le surveiller :
  - consomme des ressources (réseau, espace disque)
  - temps de mise en place et de maintenance

- De nos jours on aurait tendance à tout superviser et utiliser une AI pour faire le tri.

Méthodologie

Faire - Do

- Mise en place de l'infrastructure et des procédures définies dans l'étape de planification :
  - Utilisation d'outils de collectes d'informations (SNMP, agents logiciels sur les serveurs, les équipements réseaux ...)
  - Centralisation et analyse des fichiers log (historique d'événements)
  - Envoie de message d'alerte (Mail, SMS ...)
  - Automatisation du traitement de l'information : face à l'importance (taille et criticité) des réseaux actuels==> difficile de prendre connaissance de toutes les informations et de réagir pro-activement ==> nécessité d'automatiser l'analyse des infos remontées
  - Développement de scripts et d'interfaces.
  - Mise en place de solution de télémétrie (Telemetry). Push model.

- Mise en place de statistiques et de procédures pour la résolution des problèmes récurrents

- Nécessité de mettre en place des méthodes de mesure fiables et objectives (cf métrologie).

- Reporting : présenter les données dans des formats facilement exploitables : graphiques. (tableaux de chiffres peu attrayants et parfois illisibles).

Vérifier - Check (travail de bac +5)

- Est-ce que les choix (indicateurs, métriques, seuils, outils ...) permettent de mettre en évidence les problèmes ?

- Si un incident se produit ? A-t-on été prévenu avant l'incident ? A t-on eu le temps de traiter le problème avant qu'il n'impacte les utilisateurs ou qu'il immobilise l'entreprise ?

Corriger - Act

- Prévoir les corrections nécessaires dans les indicateur, les mesures les méthodes d'alertes ....

Contraintes

- La supervision doit s'adapter à des matériels et logiciels hétérogènes (en informatique : adoption rapide de nouvelles technologies ==> milieux très hétérogènes).

- La mise en place d'une solution de supervision est un projet à part entière :
  - doit se faire de manière progressive.
  - faire accepter l'outil : aussi complexe que de le mettre en place.
  - Si l'administrateur se borne à vouloir considérer de projet d'un point de vue purement technique, il est voué à l'échec. Exemple : les autres administrateurs peuvent mal vivre l'arrivée de la solution.

Tour d'horizon des outils pour la supervision

- Deux grandes catégories d'outils :
  - La supervision réseau : orientée métrologie, qualité de service
  - La supervision des services : disponibilité d'un service, temps de réaction
  - Tendance actuelle : un seul outil faisant tout.

- Deux façons de fonctionner :
  - Utilisation d'outils disparates
  - Agrégation de toutes les informations dans un seul outil

- Généralement : le serveur de supervision fait du « polling ».

Définir le terme polling

- Nouveau mode de fonctionnement : Telemetry. Mode Push : équipement ou service envoie les donnée (métriques) directement sans avoir besoin de requête de la part du serveur de supervision.

Tour d'horizon des outils pour la supervision

- Supervision de l'infrastructure réseau :
  - Logiciels de diagnostics standards : Wireshark, tshark (port mirroring), ping, traceroute, iperf, protocoles propriétaires ex Cisco CDP (Cisco Discovery Protocol)…
  - SNMP (Simple Network Management Protocol)
    - Collecter des données
    - Faire remonter des alarmes (trap)
  - Développement de scripts spécifiques
  - Logiciel d'analyse de flux :
    - Connaître les usages et applications du réseau – la nature du trafic.
    - Statistique d'utilisation par protocole, port, préfixe IP.
    - Classement des machines les + gourmandes du réseau.
    - Surveiller l'utilisation des classes de service.
    - Netflow (IP Flow Information Export)-sflow

- Outils de la supervision des serveurs et services :
  - disponibilité de la machine hébergeant le service et disponibilité du service
  - Scripts personnalisés permettant de mesurer les temps de réponses http, smb, ldap ...
  - Installation d'agents sur les machines supervisées.
  - Centralisation et outils d'analyses des logs

- Exemples de logiciels intégrés
  - Open source : Nagios / Centreon / Eyes of Network / Zabbix
  - HP OpenView, actuellement IT Performance Suite

Le protocole SNMP

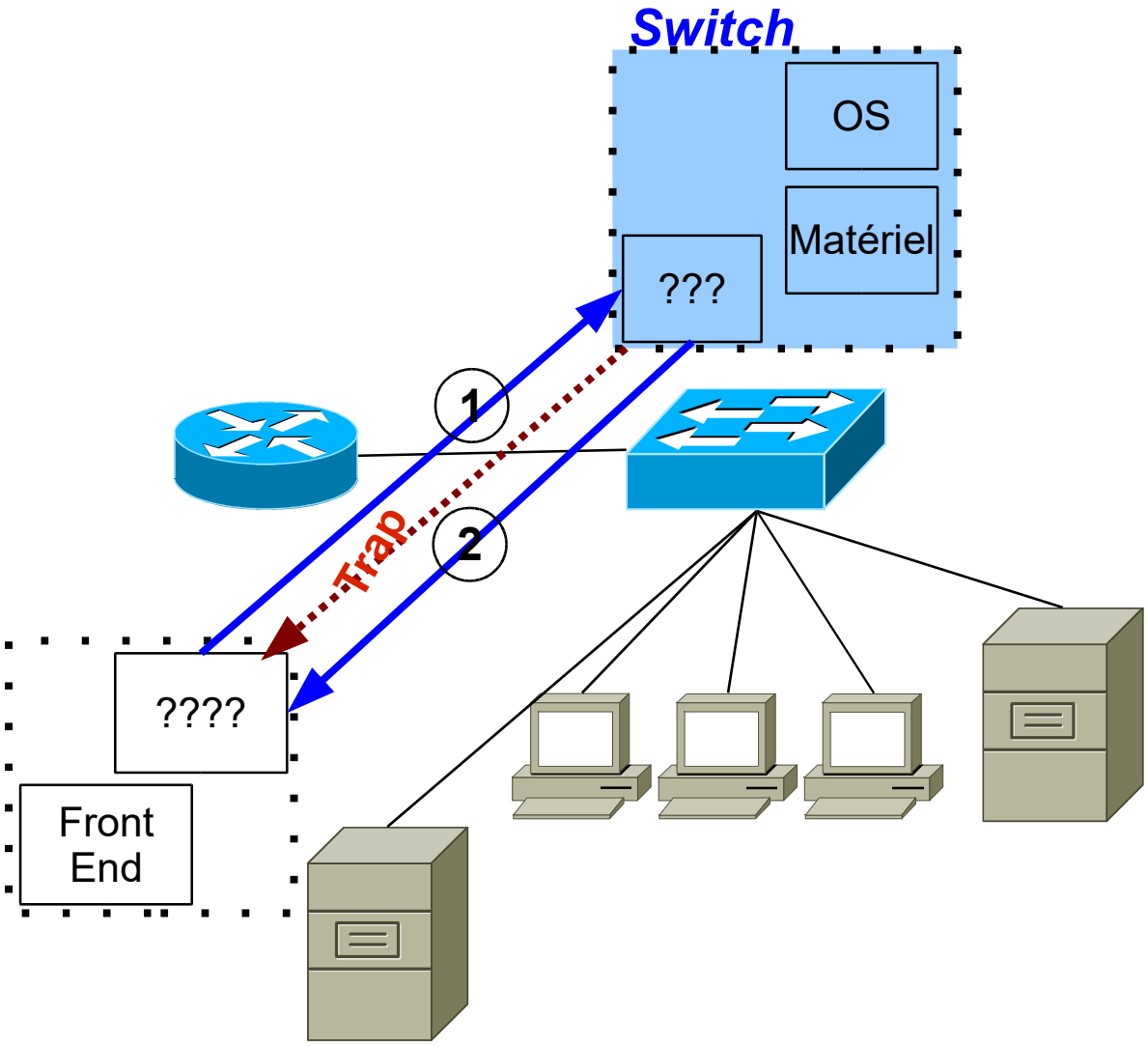
- SNMP (Simple Network Management Protocol).

- Trois versions de SNMP : version 1, 2 (2c) et 3. Actuellement version la plus utilisée : version 2c. Principal inconvénient : mot de passe en clair.

- Version 3 authentification et confidentialité.

- Trois fonctions associées à SNMP :
  - accéder (à distance) à des indicateurs de bon fonctionnement de l'équipement : temps écoulé depuis la mise en route, nombre d'octets ayant transité par une interface, charge CPU, @IP des interfaces, table arp ...
  - envoie de message d'alarme (trap)
  - modifier (à distance) la configuration de l'équipement : changer l'adresse IP, ajouter une route dans la table de routage …

- Dans SNMP il y a 2 acteurs :
  - Une station de supervision Network Management Station :
    - Interroge (requête snmp) à intervalles réguliers des équipements réseaux
    - Réceptionne des alarmes (trap)
  - Un agent : logiciel fonctionnant dans les équipements réseaux (commutateur, routeur, serveur ...) qui :
    - interroge les objets (matériels - interface, processeur, ram ... et logiciels systèmes d'exploitations, processus en cours de fonctionnement) définis par une base (MIB)
    - met à jour ces informations (compteur)
    - renvoie ces informations lorsque le NMS lui demande
    - envoie des alertes directement au NMS.



?.....

1 :
2 :
3 :

- Protocoles de transport
  - UDP :
    - Agent écoute sur le port 161 : réception de requêtes
    - NMS écoute sur le port 162 ; réception de trap

Justification du choix de UDP

- Avantages :

- Inconvénients :

Sécurité SNMPv1 et v2

- Seul processus de sécurité : identification par mot de passe . Le mot de passe dans le jargon SNMP « community » : communauté.

- Deux actions possibles selon le mot de passe (communauté) transmis. :
  - Read
  - Read/Write

SNMP - Sécurité

- Par défaut dans de nombreux équipements réseaux :
  - communauté « public » --> lire
  - communauté « private » --> lire et écrire

- Attention : changer ces valeurs avant la mise en production d'un équipement.

- Possibilité de restreindre l'accès en fonction de l'@IP sce (@IP du NMS).

- ATTENTION : la communauté circule en clair sur le réseau (en snmp version 2 et inférieure).

- Que faut-il faire pour éviter qu'un utilisateur mal intentionné capture la « communauté » ? Illustrer par un schéma.

SNMPv3 : authentification et confidentialité. Cf exercice de TD.

SNMP – La MIB

- SNMP : accéder (à distance) à des indicateurs de bon fonctionnement de l'équipement :
  - temps écoulé depuis la mise en route
  - nombre d'octets ayant transité par une interface,
  - charge CPU,
  - @IP des interfaces
  - table arp ...

- Dans SNMP **variables** sont des « **objets** ».
  - Exemple : le nombre d'octets sortant d'une interface : un objet.

- Problèmes :
  - Comment savoir quel objet est accessible dans un équipement ?
  - Comment spécifier quel objet on demande ?
  - Quel est le format de codage de la valeur associée à l'objet : chaînes de caractères, entiers, réels ?
  - Comment gérer l'hétérogénéité des équipements et des logiciels ?
  - Nécessité que les objets soient définis d'une façon standard et indépendante des constructeurs

- Solution : La MIB (Management Information Base)

- MIB : Base de données normalisée qui définit tous les objets possibles (variables)
  - répertoriés par les organismes de standardisation (IETF, IANA …).
  - qui pourront être accédés via le protocole SNMP.

SNMP – La MIB

- Pour résumer : SNMP = ensemble d'objets « normalisés » gérés par des agents
  - lus
  - et éventuellement modifiés par une station d'administration.

- Nécessité que les objets soient définis d'une façon standard et indépendante des constructeurs :
  - technique standard de codage de ces objets lors de leur transfert sur le réseau
  - définition d'un système identification des objets (arbre d'identification).

Les formats de données utilisés par SNMP

- Comment les données échangées sont formatées ?

- ASN-1 : Abstract Syntax Notation One (ASN.1)

- SMI : Structure of Management Information (SMI)

- BER : Basic Encoding Rules (BER) : encode les messages SNMP permettant de les transporter sur le réseau (assure l'interopérabilité Big-Little Endian …)

- Par analogie :
  - ASN-1 : syntaxe pour définition générique de type de donnée. Code lisible par humain. Code source.
  - SMI : rédigé en ASN-1. Rajoute des définitions/types nécessaires à la supervision. SMI a été écrit pour SNMP. Code source. Version 1 : SNMP v1, version 2 : snmp v2.
  - BER : version encodé de ASN-1. Code machine, des 0 et des 1.

ASN-1

- ASN.1 (Abstract Syntax Notation One - Notation d'extrait syntaxique).

- ASN.1 : **décrire** des structures de données complexes indépendamment d'un langage de programmation.

- Exemple :

```
CertainStructure ::= SEQUENCE {
                tag     VisibleString,
                val1    INTEGER,
                val2    INTEGER   OPTIONAL,
                reals   SET OF REAL
        }
```

  - Peut être traduit en langage C

```
typedef struct CertainStructure {
                VisibleString_t  tag;
                int              val1;
                int             *val2;        /* OPTIONAL */
                A_SET_OF(double) reals;
        } CertainStructure_t;
```

- ASN.1 notation symbolique flexible permettant de représenter
  - des types abstraits
  - des valeurs

- ASN.1 : par analogie ancêtre de xml / json (JavaScript Object Notation). Plus « performant » que xml.

- ASN.1 : syntaxe utilisée pour décrire tous les objets ainsi que leurs attributs qui peuvent être stockés dans un annuaire LDAP.

SNMP ASN-1

- ASN définit des types simples :
  - BIT STRING, an arbitrary string of bits (1 and 0).
  - IA5String, an arbitrary string of IA5 (ASCII) characters.
  - INTEGER, an arbitrary integer.
  - NULL, a null value.
  - OBJECT IDENTIFIER, an object identifier, which is a sequence of integer components that identify an object such as an algorithm or attribute type.
  - OCTET STRING, an arbitrary string of octets (eight-bit values).
  - ...

- ASN définit également des types structurés :
  - SEQUENCE, an ordered collection of **one or more** types.
  - SEQUENCE OF, an ordered collection of zero or more occurrences of a **given** type.
  - ...

- ASN.1, format également utilisé dans les annuaires LDAP

```
person OBJECT-CLASS ::= {
        SUBCLASS OF { top }
        MUST CONTAIN { commonName | surname }
        MAY CONTAIN {
         description |
         telephoneNUmber |
         userPassword |
         seeAlso
         }
        ID id-oc-person
}
```

SMI Structure of Management Information

- Rajoute des règles et des définitions à ASN.1 permettant de décrire tous les objets et tous les types nécessaire à la supervision SNMP.

- SMI utilise la syntaxe ASN-1. SMI = sous-ensemble de ASN-1.

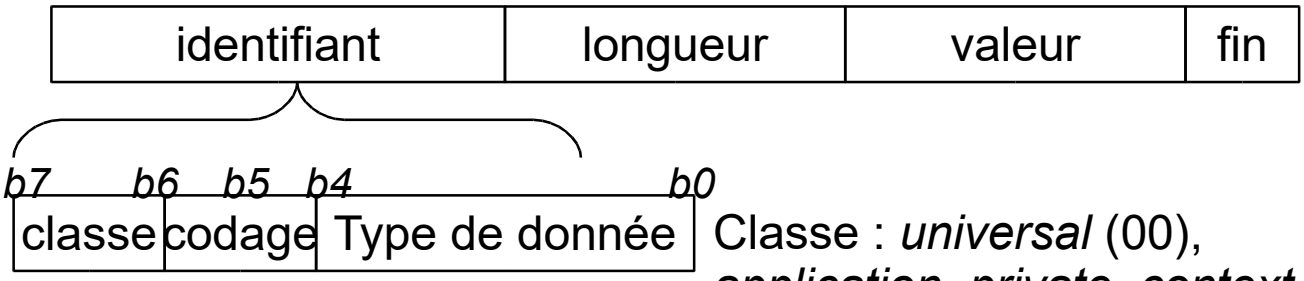
- Exemple : extrait de la RFC définissant la MIB-2 codée en SMIv2 :

```
sysContact OBJECT-TYPE
            SYNTAX  DisplayString (SIZE (0..255))
            ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
                    "The textual identification of the
                     contact person for this managed node,
                     together with informationon how to
                     contact this person."
            ::= { system 4 }
```

Nouveau type défini dans smi.

BER Basic Encoding Rules

- BER définit comment transformer en binaire/hexa un objet décrit en ASN-1.



Classe : *universal* (00), *application*, *private*, *context-specific*
Codage : primitif (0), construit (1)

Type de donnée universel

Type	Tag
INTEGER	0x02
BIT STRING	0x03
OCTET STRING	0x04
NULL	0x05
OBJECT IDENTIFIER	0x06
SEQUENCE (OF)	0x10
SET (OF)	0x11
PrintableString	0x13
T61String	0x14

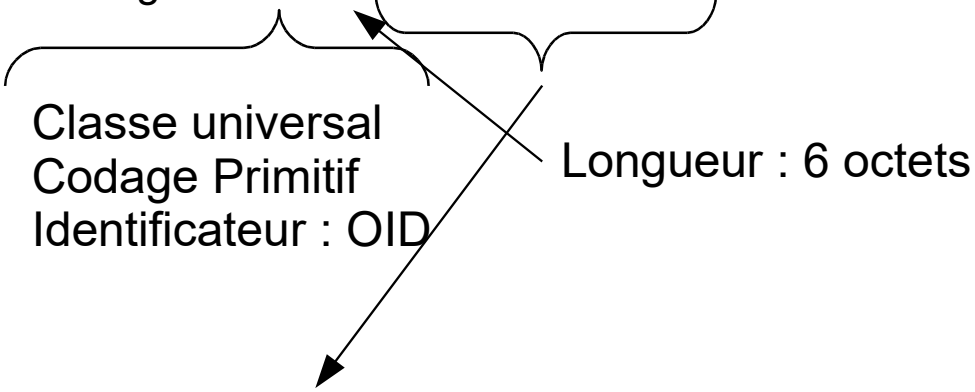
Identificateur type de donnée pour classe Application

IpAdress	0
Counter	1
Gauge	2
Time Ticks	3
Opaque	4

Avantages BER :
Encodage binaire, léger et performant (par rapport xml et json), Multiplateformes.

SNMP ASN BER

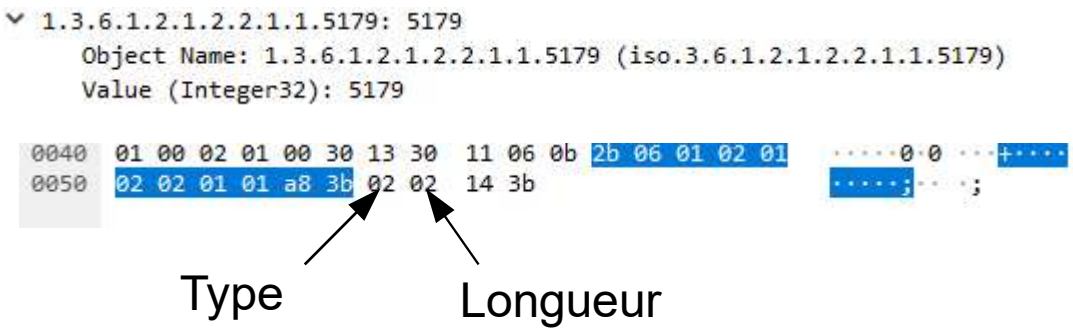
- Exemple
  - Décodage de 06 06 2B 06 01 02 01 01 05 00



2B : 43 → According to BER, the first two numbers of any OID (x.y) are encoded as one value using the formula (40*x)+y. Ici x=1, y=3.
06 → 6
01 → 1
02 → 2
01 → 1
01 → 1

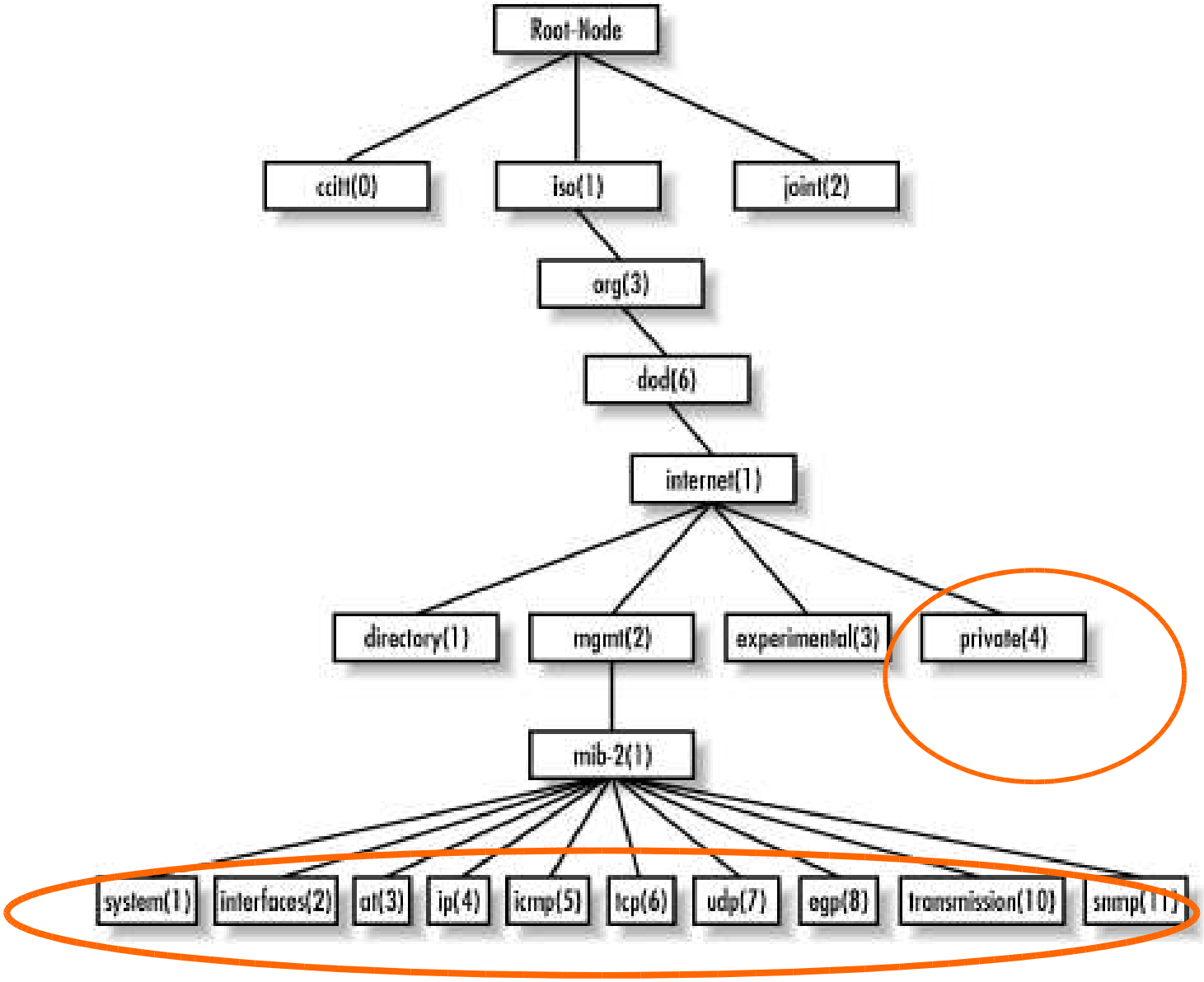
Quel est l'OID demandé ?

Autre exemple



SNMP – Espace de nommage - OID

- MIB SNMP
  - Partie de l'espace de nommage de l'ISO.
  - Structure hiérarchique : les informations sont regroupées en arbre. Chaque information a un object identifier.



Extrait de la RFC 1213

```
system          OBJECT IDENTIFIER ::= { mib-2 1 }
[...]
```

sysDescr OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters."
::= { system 1 }

sysObjectID OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMIenterprises subtree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining `what kind of box' is being managed. For example, if vendor `Flintstones, Inc.' was assigned the subtree 1.3.6.1.4.1.4242, it could assign the identifier 1.3.6.1.4.1.4242.1.1 to its `Fred Router'."
::= { system 2 }

sysUpTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time (in hundredths of a second) since the network management portion of the system was last re-initialized."
::= { system 3 }

sysContact OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION
"The textual identification of the contact person for this managed node, together with information on how to contact this person."
::= { system 4 }

Extrait de la RFC 1213

```
        sysName OBJECT-TYPE
            SYNTAX  DisplayString (SIZE (0..255))
    ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
     "An administratively-assigned name for this managed node.  By convention,
    this is the node's fully-qualified domain name."
            ::= { system 5 }

        sysLocation OBJECT-TYPE
            SYNTAX  DisplayString (SIZE (0..255))
            ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
    "The physical location of this node (e.g., `telephone closet, 3rd floor')."
            ::= { system 6 }

        sysServices OBJECT-TYPE
            SYNTAX  INTEGER (0..127)
            ACCESS  read-only
            STATUS  mandatory
            DESCRIPTION
    "A value which indicates the set of services that this entity primarily offers.

    [...]
```

- Remarque : les variables simples sont référencées en ajoutant « .0 » à l'identificateur d'objet de la variable.

- Donner l'OID de l'objet contenant le nom de la personne à contacter en cas de problème sur le matériel ?

- Donner l'OID de l'objet contenant l'emplacement où se trouve l'appareil interrogé.

SNMP - OID

- Exemple d'échanges SNMP

```
florent@portflo2:~$ snmpget -v2c -c public 192.168.137.165
1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: mau-59-srv-file
florent@portflo2:~$ snmpget -v2c -c public 192.168.137.165
sysLocation.0
SNMPv2-MIB::sysLocation.0 = STRING: Bourget du Lac Maurienne
59 59b
```

- Types de données SNMP (définis dans SMI) :
  - Types « simples » :
    - Integer, Octet String, DisplayString, Object Identifier
    - IpAddress : chaîne de 4 octets
    - PhysAddress : chaîne d'octets
    - Counter : entier non négatif dont la valeur s'accroit de 0 à 2^{32}-1 ou 2^{64} -1 puis repart à partir de 0
    - Gauge : entier dont la valeur peut augmenter ou diminuer, mais ne repart pas à 0 en atteignant son maximum. Utilisé pour la capacité d'une interface.
    - Timeticks : compteur de temps écoulé en 1/100 de secondes à partir d'un instant donné
  - Structure et tableau :
    - SEQUENCE : Ce type est comparable à une structure en langage C. **Exemple** voir page suivante : la MIB définit une SEQUENCE nommée UdpEntry qui contient une description des points de terminaison UDP actifs d'un agent. La structure possède 2 « variables » :

```
udpLocalAddress
udpLocalPort
```

    - SEQUENCE OF : C'est la définition d'un vecteur (tableau à une colonne), dont tous les éléments ont le même type de données.

      Si chaque élément possède un type de données simples (ex INTEGER), alors il s'agit d'un vecteur simple (tableau unidimensionnel). SNMP utilise aussi ce type avec chaque élément du vecteur dont la structure est en SEQUENCE. Nous pouvons l'assimiler à un tableau à 2 dimensions.

Exemple de fonctionnement de la MIB - UDP

- Extraits de la RFC 1213.

```
The objects defined in MIB-II have the OBJECT IDENTIFIER
prefix:
      mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }
[...]
udp           OBJECT IDENTIFIER ::= { mib-2 7 }
[...]
-- the UDP group
-- Implementation of the UDP group is mandatory for all
-- systems which implement the UDP.

          udpInDatagrams OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
"The total number of UDP datagrams delivered to UDP users."
              ::= { udp 1 }

          udpNoPorts OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
"The total number of received UDP datagrams for which there
was no application at the destination port."
              ::= { udp 2 }

          udpInErrors OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
"The number of received UDP datagrams that could not be
delivered for reasons other than the lack of an application
at the destination port."
              ::= { udp 3 }
```

MIB - UDP

```
udpOutDatagrams OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
"The total number of UDP datagrams sent from this entity."
              ::= { udp 4 }

 -- the UDP Listener table
 -- The UDP listener table contains information about this
 -- entity's UDP end-points on which a local application is
 -- currently accepting datagrams.

          udpTable OBJECT-TYPE
              SYNTAX  SEQUENCE OF UdpEntry
              ACCESS  not-accessible
              STATUS  mandatory
              DESCRIPTION
 "A table containing UDP listener information."
              ::= { udp 5 }

          udpEntry OBJECT-TYPE
              SYNTAX  UdpEntry
              ACCESS  not-accessible
              STATUS  mandatory
              DESCRIPTION
 "Information about a particular current UDP listener."
              INDEX   { udpLocalAddress, udpLocalPort }
              ::= { udpTable 1 }

          UdpEntry ::=
              SEQUENCE {
                  udpLocalAddress
                      IpAddress,
                  udpLocalPort
                      INTEGER (0..65535)
              }
```

MIB - UDP

```
udpLocalAddress OBJECT-TYPE
                SYNTAX  IpAddress
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
"The local IP address for this UDP listener.  In the case of
a UDP listener which is willing to accept datagrams for any
IP interface associated with the node, the value 0.0.0.0 is
used."
                ::= { udpEntry 1 }

           udpLocalPort OBJECT-TYPE
                SYNTAX  INTEGER (0..65535)
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
 "The local port number for this UDP listener."
                ::= { udpEntry 2 }
```

- Quelle est selon vous la signification de ::= { udp X } avec X un nombre qui se trouve à la fin de chaque définition d'objet ?

- Dessiner la partie de l'arbre représentant les objets accessibles via SNMP à partir de MIB-2.UDP.

MIB - UDP

- Quelle commande snmpget doit-on saisir pour demander le nombre de datagrammes entrant à la machine 192.168.137.165 ?

- On s'intéresse maintenant à la représentation de la table **udptable**.
  - Quelle est la « syntaxe » associée à udpTable ? A partir de cette syntaxe, donner une première représentation de udpTable.

  - Qu'est-ce qu'un udpEntry ? En utilisant la syntaxe de udptable et de udpEntry donner une représentation détaillée udpTable.

  - Un index est défini pour un objet de type udpEntry, de quoi est composé cet index ?

MIB - UDP

- On a dans un système la table suivante :

udpLocalAddress	*udpLocalPort*
0.0.0.0	67
0.0.0.0	161
0.0.0.0	520

- Donner la valeur de l'OID permettant d'accéder à la première case du tableau. Quelle sera la valeur si on fait un GET sur cet OID ?

- Donner la valeur de l'OID permettant d'accéder à la deuxième case du tableau (udpLocalPort).

Donner les valeurs renvoyées pour des requêtes sur les objets suivants :

Objet	**Valeur**
1.3.6.1.2.1.7.5.1.2.0.0.0.0.161	
1.3.6.1.2.1.7.5.1.1.0.0.0.0.520	

Les opérations SNMP

- Get : permet de récupérer la valeur d'un OID
  - 2 trames :
    - get-request --> demande la valeur d'un objet (envoyé par le NMS)
    - get-response --> envoie de la valeur de l'objet.

- Get-next
  - Permet de récupérer la valeur du prochain OID dans l'arbre de la MIB.
  - Exemple :

get-next 1.3.6.1.2.1
Renvoie le prochain OID valable soit :
1.3.6.1.2.1.1.1.0 (system.sysdescr.0)

    - Si on fait un get-next 1.3.6.1.2.1.1.1.0, donner la valeur de l'oid de l'objet qui sera envoyé.

- Get-bulk (voir documentation).

- Set : change la valeur d'un OID. Uniquement possible si accès en lecture/ecriture.

- SNMP Inform : utilisé pour la communication entre NMS. Si plusieurs NMS mis en place possibilité de « faire suivre » le message en utilisant un SNMP inform.

Les commandes SNMP

- Exemple **Get-bulk**

```
-Cn<NUM> : Set the non-repeaters field in the GETBULK PDU.
This specifies the  number  of  supplied variables that
should not be iterated over.  The default is 0.

-Cr<NUM> : Set the max-repetitions field in the GETBULK PDU.
 This  specifies  the maximum number of iterations over the
repeating variables.  The default is 10.

florent@portflo2:~$ snmpbulkget -v2c -Cn2 -Cr10 -c public
192.168.137.165 system udp ip ifTable
SNMPv2-MIB::sysDescr.0 = STRING: Linux mau-59-srv-file
2.6.18-194.el5 #1 SMP Fri Apr 2 14:58:35 EDT 2010 i686
UDP-MIB::udpInDatagrams.0 = Counter32: 8753
IP-MIB::ipForwarding.0 = INTEGER: notForwarding(2)
IF-MIB::ifIndex.1 = INTEGER: 1
IP-MIB::ipDefaultTTL.0 = INTEGER: 64
IF-MIB::ifIndex.2 = INTEGER: 2
IP-MIB::ipInReceives.0 = Counter32: 389477
IF-MIB::ifIndex.3 = INTEGER: 3
IP-MIB::ipInHdrErrors.0 = Counter32: 0
IF-MIB::ifIndex.4 = INTEGER: 4
IP-MIB::ipInAddrErrors.0 = Counter32: 108674
IF-MIB::ifIndex.5 = INTEGER: 5
IP-MIB::ipForwDatagrams.0 = Counter32: 0
IF-MIB::ifDescr.1 = STRING: lo
IP-MIB::ipInUnknownProtos.0 = Counter32: 0
IF-MIB::ifDescr.2 = STRING: eth1
IP-MIB::ipInDiscards.0 = Counter32: 0
IF-MIB::ifDescr.3 = STRING: eth2
IP-MIB::ipInDelivers.0 = Counter32: 195177
IF-MIB::ifDescr.4 = STRING: eth0
IP-MIB::ipOutRequests.0 = Counter32: 132744
IF-MIB::ifDescr.5 = STRING: sit0
```

- *Analysez la capture ci-dessus.*

Les Traps SNMP

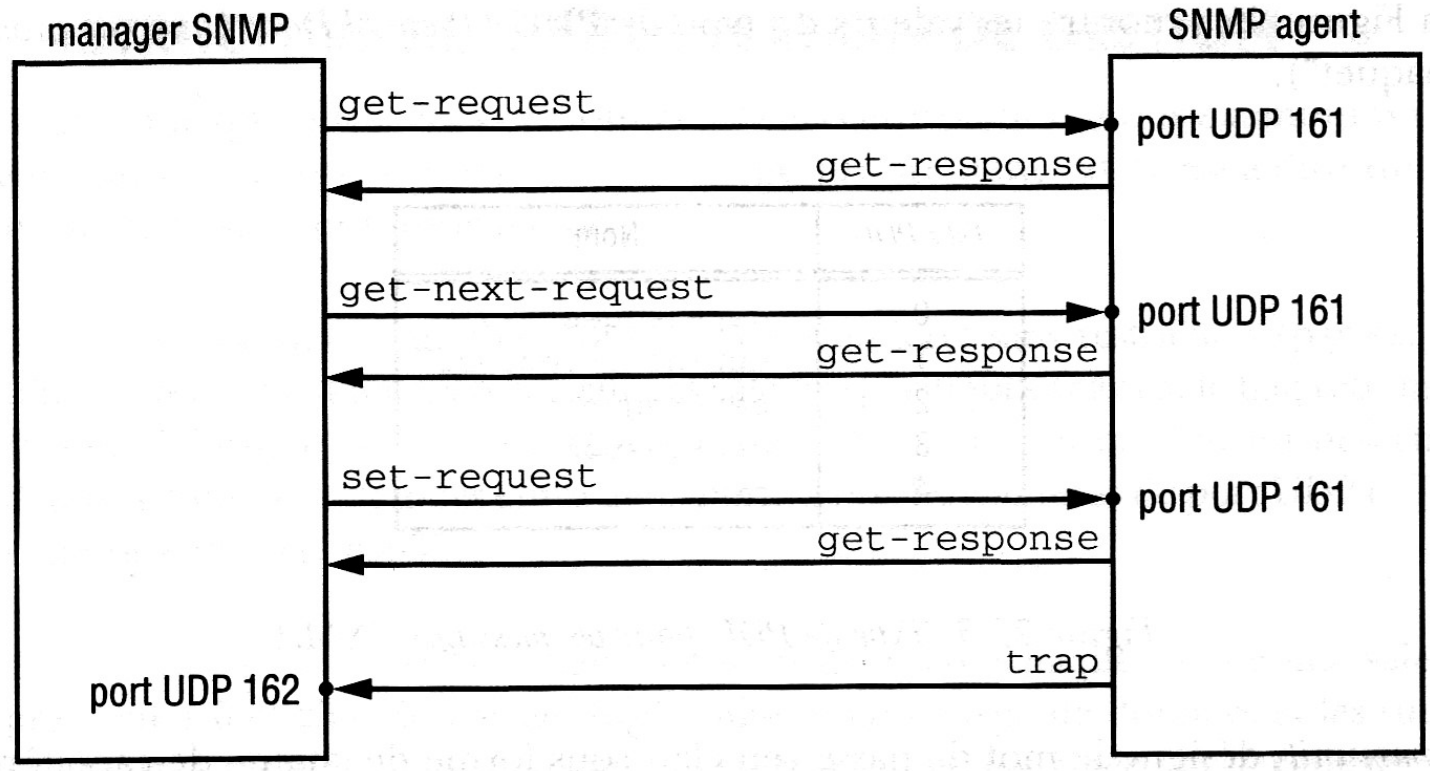
- Trap (piège) --> message d'alarme

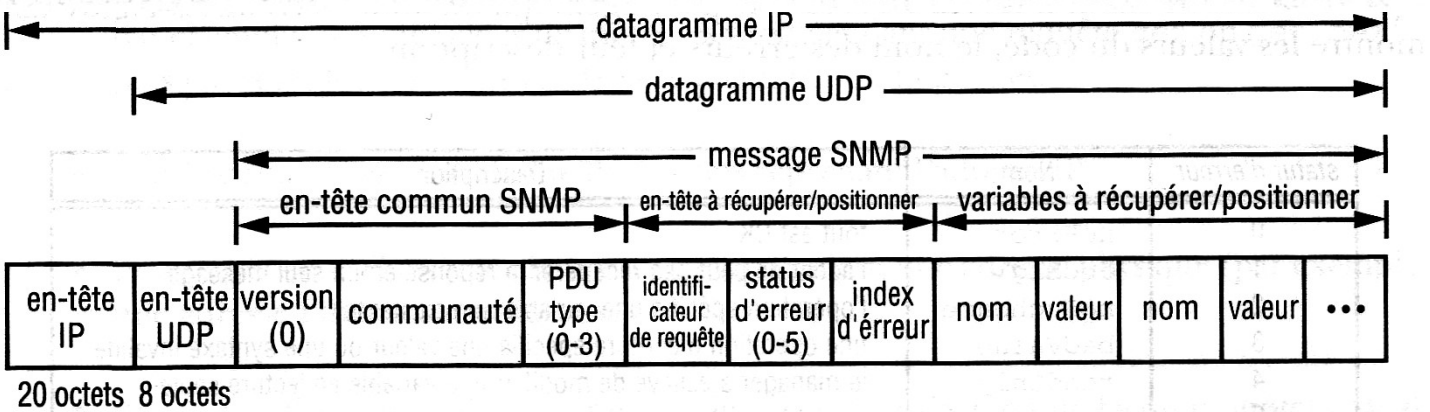
- Envoyé par l'agent à destination du NMS. L'agent est configuré avec l'@IP du NMS à qui il doit envoyer le trap.

- Un numéro contenu dans le message permet d'identifier la cause de l'alarme : 7 alarmes génériques ont été initialement définies.
  - Coldstart (0) : L'agent a redémarré. Toutes les OID sont réinitialisés à 0. Permet de détecter lorsqu'une machine a été éteinte.
  - Warmstart (1) : L'agent s'est auto redémarré. (Mauvaise configuration de l'agent ou de SNMP).
  - Linkdown (2) : Une interface devient inactive
  - Linkup(3) : Une interface devient active
  - AuthenticationFailure (4) : Mauvaise communauté envoyée à l'agent.
  - EGPNeighborLoss (5) Un voisin EGP n'est plus joignable
  - EntrepriseSpecific(6) Définit par le fabricant.

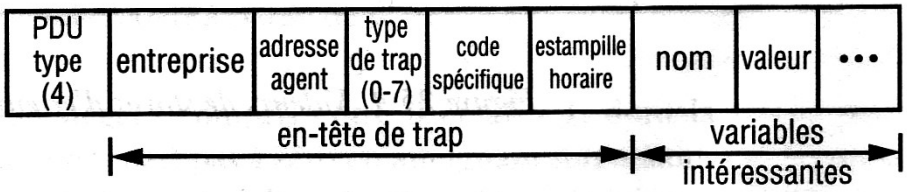
- Informations contenues dans le message d'alerte sous la forme d'OID.

Les commandes SNMP- Résumé





PDU type	Name
0	get-request
1	get-next-request
2	get-response
3	set-request
4	trap




```
▷ User Datagram Protocol, Src Port: 36199 (36199), Dst Port: snmp (161)
▽ Simple Network Management Protocol
    version: v2c (1)
    community: public
  ▽ data: get-next-request (1)
    ▽ get-next-request
        request-id: 265644651
        error-status: noError (0)
        error-index: 0
      ▽ variable-bindings: 1 item
        ▽ 1.3.6.1.2.1.1.1.0: Value (Null)
            Object Name: 1.3.6.1.2.1.1.1.0 (iso.3.6.1.2.1.1.1.0)
          ▽ Value (Null)
            ▷ [Expert Info (Note/Undecoded): Unresolved value, Missing MIB]
```