

Gestion de réseaux informatiques (GRX)

Chapitre 3: SNMP

Stephan Robert



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD

Introduction

Avec SNMP nous pouvons entre autres:

- Changer l'état d'une carte réseau
- Observer la vitesse à laquelle une interface fonctionne
- Monitorer la température d'un switch

SNMP était prévu pour gérer les routeurs principalement mais il fonctionne aussi avec d'autres types d'équipements: switches, hosts (Windows et Linux), imprimantes, équipements électriques, ...

Que se passe-t-il si un serveur tombe le vendredi après-midi alors que tout le monde est parti en week-end, y compris les administrateurs système? SNMP permet de faire le monitoring constant du réseau.



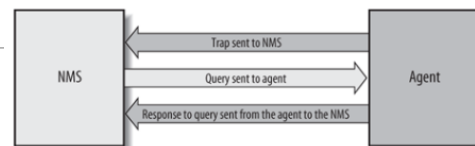
RFCs et versions de SNMP

- SNMPv1, RFC 1157. La sécurité est basée sur le concept de « communauté » (mot-de-passe pour accéder aux informations). Communautés typiques de SNMPv1: read-only, read-write, trap.
- SNMPv2 dont les communautés se réfèrent à la version 2. RFC 3416, RFC 3417, RFC 3418.
- SNMPv3: dernière version de SNMP. Principale contribution: sécurité. RFC 3410, RFC 3411, RFC 3412, RFC 3413, RFC 3414, RFC 3415, RFC 3416, RFC 3417, RFC 3418, RFC 2576.

3

Agents et managers

Protocole de communication et architecture client-serveur.



- Un manager SNMP (NMS: Network Management Station) est un software qui est installé sur un ordinateur, qui est capable d'effectuer des tâches de gestion du réseau. Responsable du polling et reçoit des trapes (envoyées de façon asynchrone) des agents.
- Un agent, software (daemon sous Linux) qui est installé sur le matériel à observer. Il est courant d'avoir des agents SNMP installés sur les équipements.
- Protocole SNMP pour la communication entre les agents et les managers (ports 161 et 162)

4

Concepts (FCAPS)

- **F. Fault Management.** (isolation du problème, résolution du problème, enregistrement du problème et de sa résolution)
- **C. Configuration Management.** Gestion de la configuration (version d'un OS, nombre d'interfaces réseau + vitesse, nombre de disques, nombre de CPUs, RAM,...)
- **A. Accounting Management.** partage équitable des ressources entre utilisateurs.
- **P. Performance Management.** Gestion de la performance (données échangées, seuils à ne pas dépasser pour certains flots de données, ...)
- **S. Security Management.** Gestion de la sécurité (Pare-feux, systèmes d'intrusion (Intrusion Detection systems, IDS, Intrusion Prevention Systems, IPS), systèmes anti-virus, gestion de règles établies). Gestion des utilisateurs (droits aux ressources, à la LB)

5

Concepts (cont...)

Niveaux d'activité

1. Inactif. Aucun monitoring. On ignore les alarmes.
2. Réactif. Aucun monitoring. On réagit en cas de problème.
3. Interactif. Monitoring et dépannage interactif.
4. Proactif. Monitoring des équipements. Processus de restauration automatique

Report des analyses de tendance

Corrélation des alarmes. Analyse des causes. Dépannage sur la base de messages qui ne sont pas toujours clairs



6

Changer?

- Planification pour un changement (- calculer et analyser les risques, - documenter les risques potentiels liés aux: changements de topologie, changements de configuration, changement de routes,..., - créer un template pour le déploiement affectant plusieurs sites, - documenter le réseau, les versions de software, le design du réseau en détail, DNS,...)
- Gérer le changement (avoir des rendez-vous périodiques, documenter les exigences des clients, l'impact sur le business, plan de sauvetage, plan d'adressage, ...)

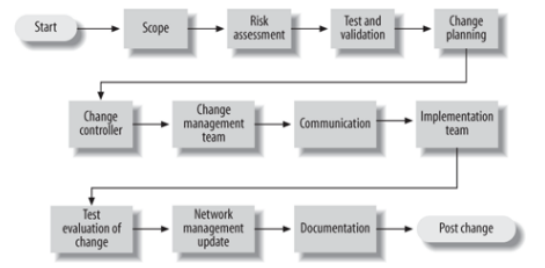
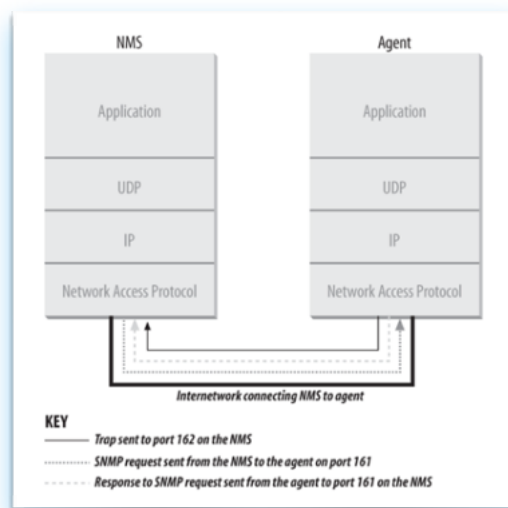


Figure 1-2. Process flow for planned change management

7

SNMP et UDP

- UDP car sans connexion et peu d'overhead (-> à SNMP de gérer)
- Le manque de fiabilité d'UDP ne présente aucun problème. Au pire NMS ne reçoit aucune réponse. Pour les trapes NMS ne saura jamais qu'elle a été envoyée...
- Port 161 pour envoyer et recevoir des requêtes. Port 162 pour recevoir des trapes.



8

Communautés (*communities*) SNMP

Etablissement d'une relation de confiance entre les managers et les clients. Trois communautés: *read-only*, *read-write*, *trap*. Une communauté est un mot-de-passe comme on en a un sur notre ordinateur.

- Read-only: lecture de valeurs sans modification de paramètres
- Read-write: lecture et modification possible de paramètres (désactivation d'interfaces, remise de compteurs à zéro, changement de la configuration d'un routeur,...)
- Trapes: alertes des clients.

Community: typiquement « public » pour read-only et « private » pour read-write. CHANGER ces valeurs avant de mettre votre réseau en production! Utiliser les mêmes règles qu'avec les mots-de-passe Linux!

9

Structure de la MIB

Structure Management Information Version1 (SMIv1), RFC 1155.

Trois attributs:

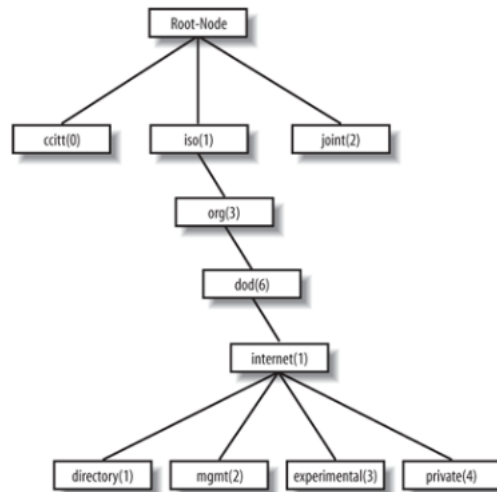
- **Nom:** Object Identifier (OID) va définir de manière unique un objet géré, sous forme lisible
- **Type et syntaxe:** Représentation des données échangées entre le manager et les clients. Sous-ensemble de Abstract Syntax Notation (ASN.1). Indépendant des machines (un PC-Win8 peut communiquer avec une SPARC sans se préoccuper de l'ordre des octets)
- **Codage:** Une instance d'un objet géré est codé dans une chaîne de caractères d'octets en utilisant Basic Encoding Rules (BER). Ensuite peut être transporté par Ethernet par exemple.

10

Noms des OIDs



- Objets gérés: organisées en hiérarchie sous forme d'arbre.
- Série de nombres séparés par un « . »
- Sommet de l'arbre: racine (root).
iso(1): sous-arbre (subtree). Sans enfants: feuille (*ccitt(0)*, *joint(2)*).
 On va se concentrer sur *iso(1).org(3).dod(6).internet(1)* ou *1.3.6.1*.



11

Noms des OIDs (cont...)

Définition d'internet avec ses sous-arbres:

```

internet    OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
directory   OBJECT IDENTIFIER ::= { internet 1 }
mgmt        OBJECT IDENTIFIER ::= { internet 2 }
experimental OBJECT IDENTIFIER ::= { internet 3 }
private     OBJECT IDENTIFIER ::= { internet 4 }
  
```

`::=` est l'opérateur « définition »

Internet Assigned Numbers Authority (IANA): gère les noms des individus, entreprises, institutions, organisations, ... Cisco a le no 9 par exemple (private enterprise).

12

SMIv1, types de données

Datatype	Description
INTEGER	A 32-bit number often used to specify enumerated types within the context of a single managed object. For example, the operational status of a router interface can be <i>up</i> , <i>down</i> , or <i>testing</i> . With enumerated types, 1 would represent up, 2 down, and 3 testing. The value zero (0) must not be used as an enumerated type, according to RFC 1155.
OCTET STRING	A string of zero or more octets (more commonly known as bytes) generally used to represent text strings, but also sometimes used to represent physical addresses.
Counter	A 32-bit number with minimum value 0 and maximum value $2^{32} - 1$ (4,294,967,295). When the maximum value is reached, it wraps back to zero and starts over. It's primarily used to track information such as the number of octets sent and received on an interface or the number of errors and discards seen on an interface. A Counter is monotonically increasing, in that its values should never decrease during normal operation. When an agent is rebooted, all Counter values should be set to zero. Deltas are used to determine if anything useful can be said for successive queries of Counter values. A delta is computed by querying a Counter at least twice in a row and taking the difference between the query results over some time interval.
OBJECT IDENTIFIER	A dotted-decimal string that represents a managed object within the object tree. For example, <i>1.3.6.1.4.1.9</i> represents Cisco Systems' private enterprise OID.
NULL	Not currently used in SNMP.
SEQUENCE	Defines lists that contain zero or more other ASN.1 datatypes.
SEQUENCE OF	Defines a managed object that is made up of a SEQUENCE of ASN.1 types.
IpAddress	Represents a 32-bit IPv4 address. Neither SMIv1 nor SMIv2 discusses 128-bit IPv6 addresses .

13

SMIv1, types de données, cont...

IpAddress	Represents a 32-bit IPv4 address. Neither SMIv1 nor SMIv2 discusses 128-bit IPv6 addresses .
NetworkAddress	Same as the IpAddress type, but can represent different network address types.
Gauge	A 32-bit number with minimum value 0 and maximum value $2^{32} - 1$ (4,294,967,295). Unlike a Counter, a Gauge can increase and decrease at will, but it can never exceed its maximum value. The interface speed on a router is measured with a Gauge.
TimeTicks	A 32-bit number with minimum value 0 and maximum value $2^{32} - 1$ (4,294,967,295). TimeTicks measures time in hundredths of a second. Uptime on a device is measured using this datatype.
Opaque	Allows any other ASN.1 encoding to be stuffed into an OCTET STRING.

14

Exemple MIB-2

```

RFC1213-MIB DEFINITIONS ::= BEGIN

IMPORTS
    mgmt, NetworkAddress, IpAddress, Counter, Gauge,
    TimeTicks
    FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC 1212;

mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }

-- groups in MIB-II

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 }

-- the Interfaces table

-- The Interfaces table contains information on the entity's
-- interfaces. Each interface is thought of as being
-- attached to a 'subnet.' Note that this term should
-- not be confused with 'subnet,' which refers to an
-- addressing-partitioning scheme used in the Internet
-- suite of protocols.

ifTable OBJECT-TYPE
    SYNTAX SEQUENCE OF IfEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of interface entries. The number of entries
        given by the value of ifNumber."
    ::= { interfaces 2 }

IfEntry OBJECT-TYPE
    SYNTAX IfEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "An interface entry containing objects at the subnetwork
        layer and below for a particular interface."
    INDEX
        { ifIndex }
    ::= { ifTable 1 }

IfEntry ::=
    SEQUENCE {
        ifIndex
            INTEGER,
        ifDescr
            DisplayString,
        ifType
            INTEGER,
        ifMtu
            INTEGER,
        ifSpeed
            Gauge,
        ifPhysAddress
            PhysAddress,
        ifAdminStatus
            INTEGER,
        ifOperStatus
            INTEGER,
        ifLastChange
            TimeTicks,
        ifInOctets
            Counter,
        ifInUcastPkts
            Counter,
        ifInNUcastPkts
            Counter,
        ifInDiscards
            Counter,
        ifErrors
            Counter,
        ifUnknownProtos
            Counter,
        ifOutOctets
            Counter,
        ifOutUcastPkts
            Counter,
        ifOutNUcastPkts
            Counter,
        ifOutDiscards
            Counter,
        ifInErrors
            Counter,
        ifInUnknownProtos
            Counter,
        ifOutErrors
            Counter,
        ifOutQLen
            Gauge,
        ifSpecific
            OBJECT IDENTIFIER
    }

ifIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A unique value for each interface. Its value ranges
        between 1 and the value of ifNumber. The value for
        each interface must remain constant at least from one
        reinitialization of the entity's network management
        system to the next reinitialization."
    ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A textual string containing information about the
        interface. This string should include the name of
        the manufacturer, the product name, and the version
        of the hardware interface."
    ::= { ifEntry 2 }

END

```

Exemple MIB-2, cont...

- MIB importe les articles suivants de RFC 1155: `mgmt`, `NetworkAddress`, `IpAddress`, `Counter`, `Gauge`, `TimeTicks`.
- `{mib-2 2}` est équivalent à 1.3.6.1.2.1.2.
- Format d'un objet:

```

<name> OBJECT-TYPE
    SYNTAX <datatype>
    ACCESS <either read-only, read-write, write-only, or not-accessible>
    STATUS <either mandatory, optional, or obsolete>
    DESCRIPTION
        "Textual description describing this particular managed object."
    ::= { <Unique OID that defines this object> }

```

Exemple MIB-2, cont...

- Définition de ifTable:

```
ifTable
OBJECT-TYPE
SYNTAX SEQUENCE OF IfEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of interface entries. The number of entries is given by
    the value of ifNumber."
::= { interfaces 2 }
```

- avec une séquence de ifEntry:

```
IfEntry ::=
SEQUENCE {
    ifIndex
        INTEGER,
    ifDescr
        DisplayString,
    ifType
        INTEGER,
    ifMTU
        INTEGER,
    .
    .
    .
    ifSpecific
        OBJECT IDENTIFIER
}
```

17

Exemple MIB-2, cont...

- L'entrée OID de ifEntry est iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry ou 1.3.6.1.2.1.2.2.1
- L'index pour ifEntry est ifIndex (en read-only):

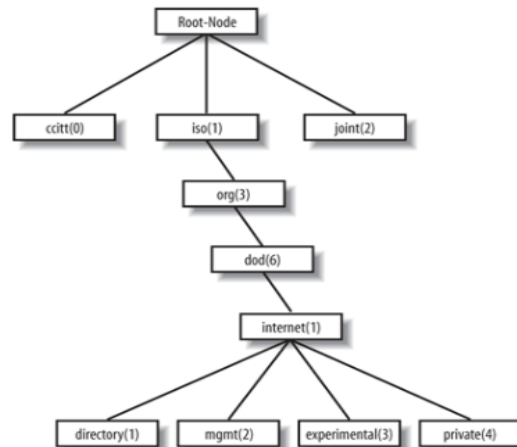
```
ifIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "A unique value for each interface. Its value ranges between
    1 and the value of ifNumber. The value for each interface
    must remain constant at least from one reinitialization of the
    entity's network management system to the next reinitialization."
::= { ifEntry 1 }
```

- Dernier objet dans la MIB: ifDescr qui est une description textuelle pour l'interface représenté par une ligne particulière dans ifTable. END: fin des fichiers de la MIB-2.

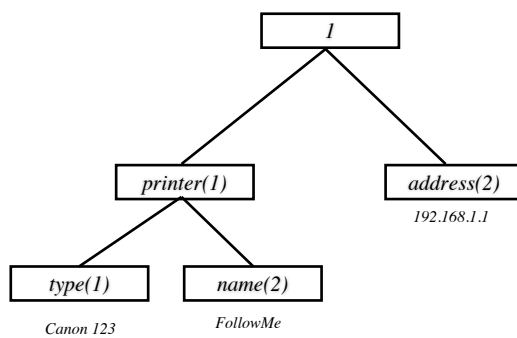
18

Exercice

- Dessinez le reste de l'arbre vous basant sur la MIB que nous venons de décrire (slide 15):

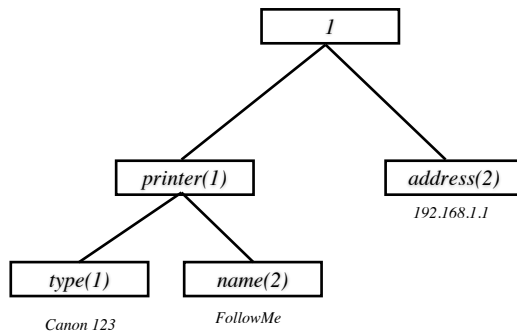


Valeurs des OIDs et des instances des objets



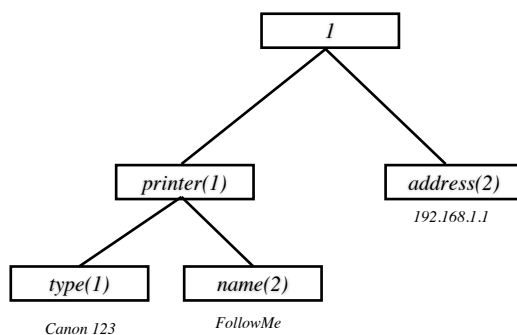
Quel est le OID de address?

Valeurs des OIDs et des instances des objets



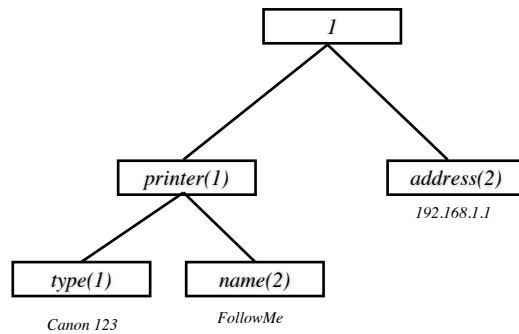
Quel est le OID de printer?

Valeurs des OIDs et des instances des objets



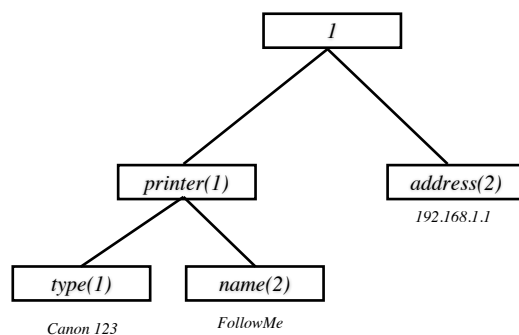
Quel est le OID de name?

Valeurs des OIDs et des instances des objets



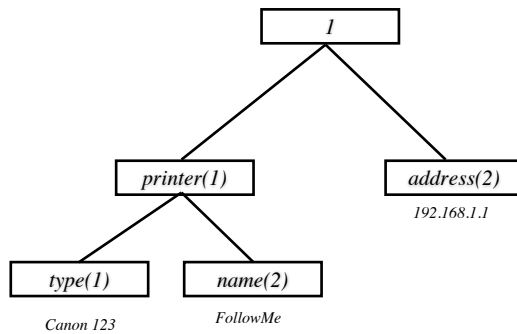
Qu'est-ce que la commande `Get(1.1.2.0)` retourne?

Valeurs des OIDs et des instances des objets



Qu'est-ce que la commande `Get(1.1.0)` retourne?

Valeurs des OIDs et des instances des objets



Qu'est-ce que la commande Get(1.1.2) retourne?

Identification des éléments d'une table

- Au tableau

Exercices

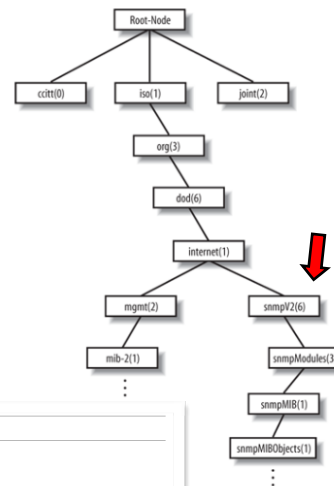
1. Quel est retourné par l'agent si une requête est faite pour 1.3.6.1.2.1.2.2.1.1?
2. Quel est retourné par l'agent si une requête est faite pour 1.3.6.1.2.1.2.2.2.3?
3. Quel est retourné par l'agent si une requête est faite pour 1.3.6.1.2.1.2.2.1.3?
4. Quel est retourné par l'agent si une requête est faite pour 1.3.6.1.2.1.2.2.1.2?
5. Quel est retourné par l'agent si une requête est faite pour 1.3.6.1.2.1.2.2.2.2?

Extensions SMIv2

- Extension de SMI. Exemple:
1.3.6.1.6.3.1.1



Datatype	Description
Integer32	Same as an INTEGER.
Counter32	Same as a Counter.
Gauge32	Same as a Gauge.
Unsigned32	Represents decimal values in the range of 0 to $2^{32} - 1$, inclusive.
Counter64	Similar to Counter32, but its maximum value is 18,446,744,073,709,551,615. Counter64 is ideal for situations in which a Counter32 may wrap back to 0 in a short amount of time.
BIT5	An enumeration of nonnegative named bits.



Définition des objets SMIv2

```

<name> OBJECT-TYPE
    SYNTAX <datatype>
    UnitsParts <Optional, see below>
    MAX-ACCESS <See below>
    STATUS <See below>
    DESCRIPTION
        "Textual description describing this particular managed object."
    AUGMENTS { <name of table> }
    ::= { <Unique OID that defines this object> }

```

Object definition enhancement	Description
UnitsParts	A textual description of the units (i.e., seconds, milliseconds, etc.) used to represent the object.
MAX-ACCESS	An OBJECT-TYPE's ACCESS can be MAX-ACCESS in SNMPv2. The valid options for MAX-ACCESS are read-only, read-write, read-create, not-accessible, and accessible-for-notify.
STATUS	This clause has been extended to allow the current, obsolete, and deprecated keywords. current in SNMPv2 is the same as mandatory in an SNMPv1 MIB.
AUGMENTS	In some cases, it is useful to add a column to an existing table. The AUGMENTS clause allows you to extend a table by adding one or more columns, represented by some other object. This clause requires the name of the table the object will augment.

30

Conventions textuelles pour SMIv2

Textual convention	Description
DisplayString	A string of NVT ASCII characters. A DisplayString can be no more than 255 characters in length.
PhysAddress	A media- or physical-level address, represented as an OCTET STRING.
MacAddress	Defines the media-access address for IEEE 802 (the standard for LANs) in canonical ^[1] order. (In everyday language, this means the Ethernet address.) This address is represented as six octets.
TruthValue	Defines both true and false Boolean values.
TestAndIncr	Used to keep two management stations from modifying the same managed object at the same time.
AutonomousType	An OID used to define a subtree with additional MIB-related definitions.
VariablePointer	A pointer to a particular object instance, such as <i>ifDescr</i> for interface 3. In this case, the VariablePointer would be the OID <i>ifDescr.3</i> .
RowPointer	A pointer to a row in a table. For example, <i>ifIndex.3</i> points to the third row in <i>ifTable</i> .

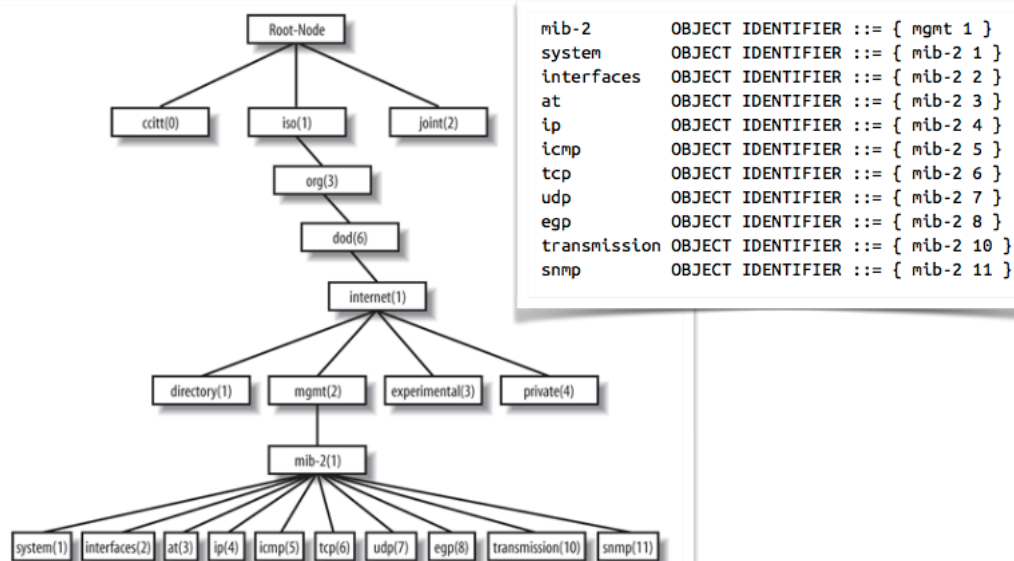
31

Conventions textuelles pour SMIv2, cont...

RowStatus	Used to manage the creation and deletion of rows in a table, since SNMP has no way of doing this via the protocol itself. RowStatus can keep track of the state of a row in a table as well as receive commands for creation and deletion of rows. This textual convention is designed to promote table integrity when more than one manager is updating rows. The following enumerated types define the commands and state variables: active(1), notInService(2), notReady(3), createAndGo(4), createAndWait(5), and destroy(6).
TimeStamp	Measures the amount of time elapsed between the device's system uptime and some event or occurrence.
TimeInterval	Measures a period of time in hundredths of a second. TimeInterval can take any integer value from 0-2147483647.
DateAndTime	An OCTET STRING used to represent date and time information.
StorageType	Defines the type of memory an agent uses. The possible values are other(1), volatile(2), nonVolatile(3), permanent(4), and readOnly(5).
TDomain	Denotes a kind of transport service.
TAddress	Denotes the transport service address. TAddress is defined to be from 1-255 octets in length.

32

MIB-2, a closer look :-)



33

Brève description des groupes MIB-2

Subtree name	OID	Description
system	1.3.6.1.2.1.1	Defines a list of objects that pertain to system operation, such as the system uptime, system contact, and system name.
interfaces	1.3.6.1.2.1.2	Keeps track of the status of each interface on a managed entity. The <i>interfaces</i> group monitors which interfaces are up or down and tracks such things as octets sent and received, errors and discards, etc.
at	1.3.6.1.2.1.3	The address translation (<i>at</i>) group is deprecated and is provided only for backward compatibility.
ip	1.3.6.1.2.1.4	Keeps track of many aspects of IP, including IP routing.
icmp	1.3.6.1.2.1.5	Tracks things such as ICMP errors, discards, etc.
tcp	1.3.6.1.2.1.6	Tracks, among other things, the state of the TCP connection (e.g., <i>closed</i> , <i>listen</i> , <i>synSent</i> , etc.).
udp	1.3.6.1.2.1.7	Tracks UDP statistics, datagrams in and out, etc.
egp	1.3.6.1.2.1.8	Tracks various statistics about the Exterior Gateway Protocol (EGP) and keeps an EGP neighbor table.
transmission	1.3.6.1.2.1.10	No objects are currently defined for this group, but other media-specific MIBs are defined using this subtree.
snmp	1.3.6.1.2.1.11	Measures the performance of the underlying SNMP implementation on the managed entity and tracks things such as the number of SNMP packets sent and received.

34

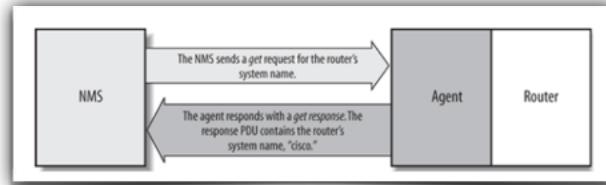
Opérations SNMP

PDU standard (format des messages échangés entre les managers et les agents)

- **get**
- getnext
- getbulk (SNMPv2, SNMPv3)
- **set**
- getresponse
- **trap**
- notification (SNMPv2, SNMPv3)
- inform (SNMPv2, SNMPv3)
- report (SNMPv2, SNMPv3)

35

Opération « get »



```
$ snmpget cisco.ora.com public .1.3.6.1.2.1.1.6.0
system.sysLocation.0 = ""
```

Attention: les commandes peuvent changer sensiblement suivant l'OS et le software que vous utilisez! Les commandes présentées ici sont celles de Net-SNMP.

```
$ snmpget -v 1 -c public 127.0.0.1 sysContact.0
```



36

Opération « get »

Trame initiée par le client

```
$ snmpget -v 1 -c public 127.0.0.1 sysContact.0
```

```

Frame 1 (85 bytes on wire, 85 bytes captured)
  Arrival Time: Sep 20, 2004 13:46:15.041115000
  Time delta from previous packet: 0.000000000 seconds
  Time since reference or first frame: 0.000000000 seconds
  Frame Number: 1
  Packet Length: 85 bytes
  Capture Length: 85 bytes
  Ethernet II, Src: 00:00:00:00:00:00, Dst: 00:00:00:00:00:00
    Destination: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Source: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Type: IP (0x0800)
  Internet Protocol, Src Addr: 127.0.0.1 (127.0.0.1), Dst Addr: 127.0.0.1 (127.0.0.1)
    Version: 4
    Total Length: 20 bytes
    Identification: 0x0000 (0)
    Flags: 0x04
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 0
    Protocol: UDP (0x11)
    Header checksum: 0x7ca4 (correct)
    Source: 127.0.0.1 (127.0.0.1)
    Destination: 127.0.0.1 (127.0.0.1)
  User Datagram Protocol, Src Port: 34066 (34066), Dst Port: snmp (161)
    Source port: 34066 (34066)
    Destination port: snmp (161)
    Length: 51
    Checksum: 0xfe46 (incorrect, should be 0xbbea)
  Simple Network Management Protocol
    Version: 1 (0)
    Community: public
    PDU type: GET (0)
    Request Id: 0x20a71b4c
    Error Status: NO ERROR (0)
    Error Index: 0
    Object identifier 1: 1.3.6.1.2.1.1.4.0 (SNMPv2-MIB::sysContact.0)
    Value: NULL
  
```

Communauté

Objet interrogé

version

Opération « get »

Réponse

Réponse sur l'objet interrogé

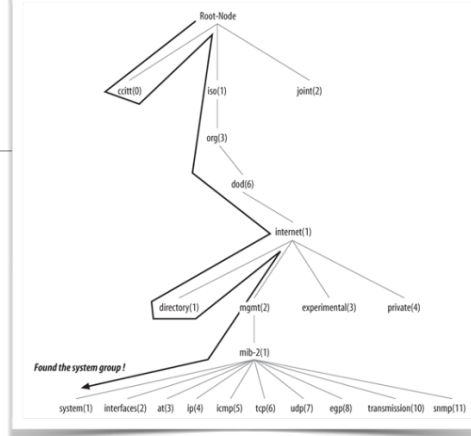
```

Frame 2 (144 bytes on wire, 144 bytes captured)
  Arrival Time: Sep 20, 2004 13:46:15.071891000
  Time delta from previous packet: 0.030776000 seconds
  Time since reference or first frame: 0.030776000 seconds
  Frame Number: 2
  Packet Length: 144 bytes
  Capture Length: 144 bytes
  Ethernet II, Src: 00:00:00:00:00:00, Dst: 00:00:00:00:00:00
    Destination: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Source: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Type: IP (0x0800)
  Internet Protocol, Src Addr: 127.0.0.1 (127.0.0.1), Dst Addr: 127.0.0.1 (127.0.0.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... 00.. = ECN-Capable Transport (ECT): 0
      .... 00.. = ECN-CE: 0
    Total Length: 130
    Identification: 0x031d (797)
    Flags: 0x04
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 0
    Protocol: UDP (0x11)
    Header checksum: 0x794c (correct)
    Source: 127.0.0.1 (127.0.0.1)
    Destination: 127.0.0.1 (127.0.0.1)
  User Datagram Protocol, Src Port: snmp (161), Dst Port: 34066 (34066)
    Source port: snmp (161)
    Destination port: 34066 (34066)
    Length: 110
    Checksum: 0xfe81 (incorrect, should be 0xdf61)
  Simple Network Management Protocol
    Version: 1 (0)
    Community: public
    PDU type: RESPONSE (2)
    Request Id: 0x20a71b4c
    Error Status: NO ERROR (0)
    Error Index: 0
    Object identifier 1: 1.3.6.1.2.1.1.4.0 (SNMPv2-MIB::sysContact.0)
    Value: STRING: Root <root@localhost> (configure /etc/snmp/snmp.local.conf)

```

Opération « getnext »

On commence par la racine et on se promène jusqu'au niveau désiré et on interroge chaque objet du groupe



```

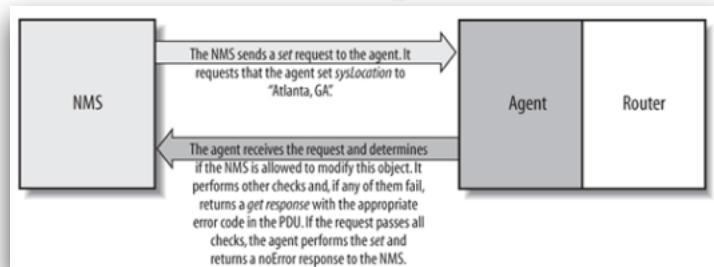
$ snmpwalk cisco.ora.com public system
system.sysDescr.0 = "Cisco IOS Software, C2600 Software (C2600-IPBASE-M),
Version 12.3(8)T3, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Tue 20-Jul-04 17:03 by eaarnas"
system.sysObjectID.0 = OID: enterprises.9.1.19
system.sysUpTime.0 = Timeticks: (27210723) 3 days, 3:35:07.23
system.sysContact.0 = ""
system.sysName.0 = "cisco.ora.com"
system.sysLocation.0 = ""
system.sysServices.0 = 6

```

39

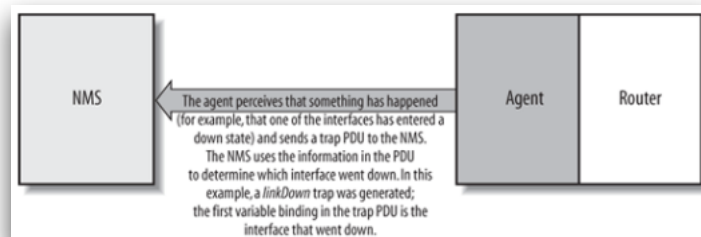
Opération « set »

```
$ snmpget cisco.ora.com public system.sysLocation.0
system.sysLocation.0 = ""
$ snmpset
cisco.ora.com private system.sysLocation.0 s "Atlanta, GA"
system.sysLocation.0 = "Atlanta, GA"
$ snmpget cisco.ora.com public system.sysLocation.0
system.sysLocation.0 = "Atlanta, GA"
```



40

Opération « trap »



- Messages envoyés d'un agent à un manager, NMS. Destination: adresse IP par exemple. Pas de ACKs.
- Messages des trapes:
 - Interface réseau qui se désactive (où est situé l'agent)
 - Interface réseau qui se réactive (où est situé l'agent)
 - Le ventilateur d'un routeur/switch qui tombe en panne
 - ... (voir slide suivante)

41

Messages génériques des trapes

Generic trap name and number	Definition
coldStart (0)	Indicates that the agent has rebooted. All management variables will be reset; specifically, Counters and Gauges will be reset to zero (0). One nice thing about the coldStart trap is that it can be used to determine when new hardware is added to the network. When a device is powered on, it sends this trap to its trap destination. If the trap destination is set correctly (i.e., to the IP address of your NMS), the NMS can receive the trap and determine whether it needs to manage the device.
warmStart (1)	Indicates that the agent has reinitialized itself. None of the management variables will be reset.
linkDown (2)	Sent when an interface on a device goes down. The first variable binding identifies the index in the <i>interfaces</i> table for the interface that went down.
linkUp (3)	Sent when an interface on a device comes back up. The first variable binding identifies which interface came back up.
authenticationFailure (4)	Indicates that someone has tried to query your agent with an incorrect community string; useful in determining if someone is trying to gain unauthorized access to one of your devices.
egpNeighborLoss (5)	Indicates that an EGP neighbor has gone down.
enterpriseSpecific (6)	Indicates that the trap is enterprise-specific. SNMP vendors and users define their own traps under the <i>private-enterprise</i> branch of the SMI <i>object</i> tree. To process this trap properly, the NMS has to decode the specific trap number that is part of the SNMP message.

42

Gestion des hosts

```

host          OBJECT IDENTIFIER ::= { mib-2 25 }

hrSystem      OBJECT IDENTIFIER ::= { host 1 }
hrStorage     OBJECT IDENTIFIER ::= { host 2 }
hrDevice      OBJECT IDENTIFIER ::= { host 3 }
hrSWRun       OBJECT IDENTIFIER ::= { host 4 }
hrSWRunPerf   OBJECT IDENTIFIER ::= { host 5 }
hrSWInstalled OBJECT IDENTIFIER ::= { host 6 }

```

OID 1.3.6.1.2.1.25 (host) comprend 6 groupes d'objets

- **hrSystem**: date du système, utilisateurs, processus
- **hrDevice** et **hrStorage**: utilisation du disque et du CPU, systèmes de fichiers
- **hrSWRun**, **hrSWRunPerf**, **hrSWInstalled**: aspects des logiciels installés sur l'ordinateur (OS, programmes fonctionnant sur la machine,...)

43

Remote Monitoring (RMON)

... que nous allons détailler plus tard....
10 groupes MIB:

rmon	OBJECT IDENTIFIER ::= { mib-2 16 }
statistics	OBJECT IDENTIFIER ::= { rmon 1 }
history	OBJECT IDENTIFIER ::= { rmon 2 }
alarm	OBJECT IDENTIFIER ::= { rmon 3 }
hosts	OBJECT IDENTIFIER ::= { rmon 4 }
hostTopN	OBJECT IDENTIFIER ::= { rmon 5 }
matrix	OBJECT IDENTIFIER ::= { rmon 6 }
filter	OBJECT IDENTIFIER ::= { rmon 7 }
capture	OBJECT IDENTIFIER ::= { rmon 8 }
event	OBJECT IDENTIFIER ::= { rmon 9 }

44

Reverse Engineering (SNMP)

Wireshark pour l'analyse des paquets si le fabricant ne publie pas ses MIBs ou si elles sont non standard.

```
Object identifier 3: 1.3.6.1.2.1.2.2.1.1 (IF-MIB::ifIndex)
Value: INTEGER: 2
Object identifier 4: 1.3.6.1.2.1.2.2.1.7 (IF-MIB::ifAdminStatus)
Value: INTEGER: up(1)
Object identifier 5: 1.3.6.1.2.1.2.2.1.8 (IF-MIB::ifOperStatus)
Value: INTEGER: up(1)
```

45

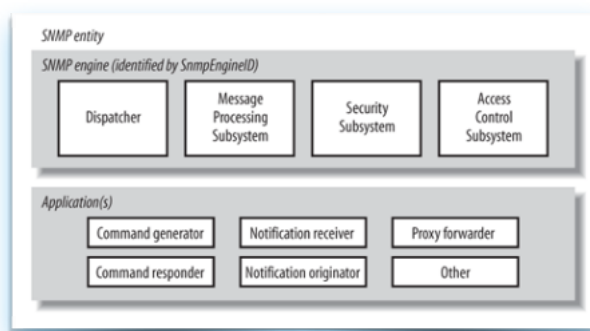
SNMPv3

- Gros défaut de SNMP dans sa version originale: sécurité. Unique protection: un mot-de-passe (community)
- La sécurité a été améliorée avec SNMPv3. Seul changement!
- Abandon de la notion d'agents et de managers. A la place: **entités SNMP**.

Number	Name
RFC 3411	Architecture for SNMP Frameworks
RFC 3412	Message Processing and Dispatching
RFC 3413	SNMP Applications
RFC 3414	User-based Security Model (USM)
RFC 3415	View-based Access Control Model (VACM)
RFC 3416	Protocol Operations for SNMPv2
RFC 3417	Transport Mappings for SNMPv2
RFC 3418	MIB for SNMPv2
RFC 2576	Coexistence Between SNMP Versions
RFC 2570	Introduction to SNMPv3
RFC 2786	Diffie-Hellman USM Key Management

SNMPv3 Engine

- Dispatcher: envoi et réception des messages
- Message Processing Subsystem: prépare les messages pour l'envoi, extrait les données des messages reçus.
- Security Subsystem: authentification (MD5 ou SHA) et sécurité (DES historiquement, 3-DES et AES)
- Access Control Subsystem: responsable pour l'accès aux objets de la MIB.



- Command generator: génère get, getnext, getbulk,...
- Command responder: répond à get, getnext, getbulk, ...
- Notification originator: génère des trapes SNMP et des notifications
- Notification originator: reçoit les trapes et informe.
- Proxy forwarder: facilite la transmission de messages entre entités

Conventions pour SNMPv3

Textual convention	Description
<code>snmpEngineID</code>	An administratively unique identifier for an SNMP engine. Objects of this type are for identification, not for addressing, even though an address can be used in the generation of a specific value. RFC 3411 provides a detailed discussion of how <code>snmpEngineIDs</code> are created.
<code>snmpSecurityModel</code>	An SNMP <code>securityModel</code> (SNMPv1, SNMPv2, or USM). USM stands for User-based Security Model, which is the security method used in SNMPv3.
<code>snmpMessageProcessingModel</code>	A message processing model used by the Message Processing Subsystem.
<code>snmpSecurityLevel</code>	The level of security at which SNMP messages can be sent, or the level of security at which operations are being processed. Possible values are <code>noAuthNoPriv</code> (without authentication and without privacy), <code>authNoPriv</code> (with authentication but without privacy), and <code>authPriv</code> (with authentication and with privacy). These three values are ordered such that <code>noAuthNoPriv</code> is less than <code>authNoPriv</code> and <code>authNoPriv</code> is less than <code>authPriv</code> .
<code>snmpAdminString</code>	An octet string containing administrative information, preferably in human-readable form. The string can be up to 255 bytes long.
<code>snmpTagValue</code>	An octet string containing a tag value. Tag values are preferably in human-readable form. According to RFC 3413, valid example tags include <code>acne</code> , <code>router</code> , and <code>host</code> .
<code>snmpTagList</code>	An octet string containing a list of tag values. Tag values are preferably in human-readable form. According to RFC 3413, valid examples of a tag list are the empty string, <code>acne router</code> , and <code>host managerStation</code> .
<code>KeyChange</code>	An object used to change authentication and privacy keys .

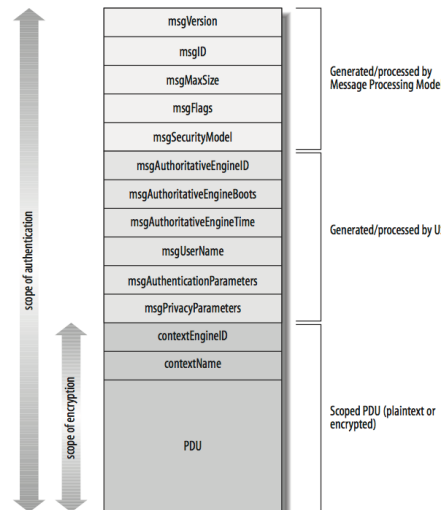
SNMPv3 Textual Conventions

48

SNMPv3

• USM: User-based Security Model

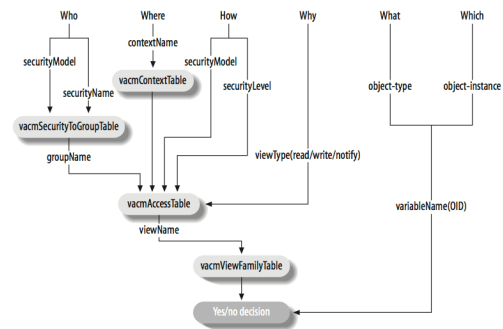
- USM a besoin que `msgSecurityParameters` contienne `snmpEngineID`, `snmpEngineBoots`, `snmpEngineTime` de l'engine qui a l'autorité. Ensuite on peut échanger des informations
- Authentication: MD5 (128 bits) ou SHA1 (160 bits)
- Privacy: CBC-DES
- Référence: "SNMPv3: A Security Enhancement for SNMP" by William Stallings.



49

SNMPv3

- VACM: View Access Control Model



- Référence: "SNMPv3: A Security Enhancement for SNMP" by William Stallings.

50

Architectures NMS (Hardware)

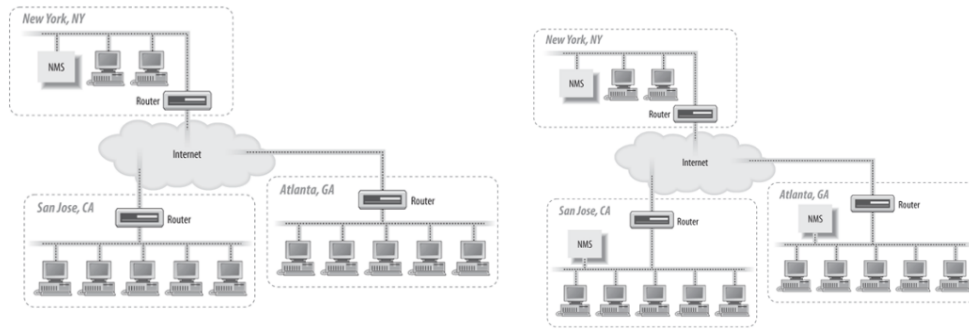
- Aucun problème avec les PC actuels: il faut un CPU 3 GHz, 1 GB de mémoire, 2 GB d'espace disque.
- Question: la quantité de données de chaque demande (get) est d'environ 1 KB. A quelle fréquence allez-vous interroger les agents (si vous en avez 1000 à surveiller)?

51

Architecture du NMS

- Un seul NMS

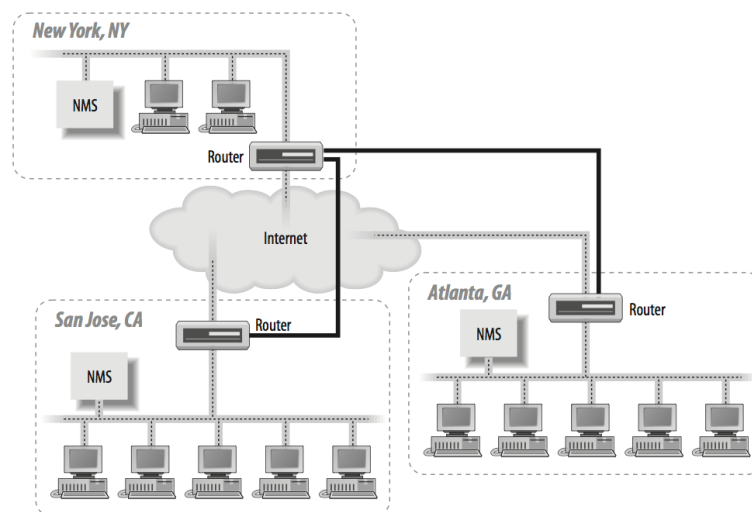
Plusieurs NMSs, architecture distribuée



52

Architecture du NMS (2)

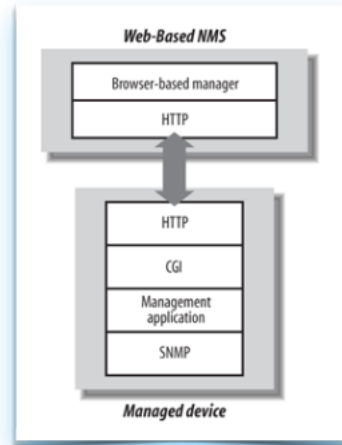
- Plusieurs



53

Management basé sur le Web

- La gestion de réseau basée sur le web élimine les logiciels traditionnels NMS.
- Utilisation de XML
- Conversion des MIBs SNMP en XML pour la portabilité
- Usage des commandes XML pour les commandes et le contrôle
- Configuration des agents et du manager: travail de laboratoire...



54

Polling, scripting (Perl) et seuils

- Récupération de valeurs depuis la MIB (script Perl, *snmpget.pl*)
- ```

#!/usr/local/bin/perl
#filename: /opt/local/perl_scripts/snmpget.pl
use BER;
use SNMP_util;
use SNMP_Session;
$MIB1 = ".1.3.6.1.2.1.1.4.0";
$HOST = "orarouter1";
($value) = &snmpget("public@$HOST", "$MIB1");
if ($value) { print "Results :$MIB1: :$value:\n"; }
else { warn "No response from host :$HOST:\n"; }

```

```

$ /opt/local/perl_scripts/snmpget.pl
Results :.1.3.6.1.2.1.1.4.0: :ORA IT Group:

```

55

## Polling, scripting (Perl)

- Récupération de valeurs depuis la MIB (script Perl, *snmpget.pl*) pour pooler n'importe quel OID

```
#!/usr/local/bin/perl
#filename: /opt/local/perl_scripts/snmpget.pl
use BER;

$MIB1 = shift;
$HOST = shift;
($MIB1) && ($HOST) || die "Usage: $0 MIB_OID HOSTNAME";
($value) = &snmpget("$HOST", "$MIB1");
if ($value) { print "Results :$MIB1: :$value:\n"; } else
{ warn "No response from host :$HOST:\n"; }

$ /opt/local/perl_scripts/snmpget.pl .1.3.6.1.2.1.1.1.0 public@orrouter1
Results :.1.3.6.1.2.1.1.1.0: : Cisco IOS Software, C2600 Software (C2600-
IPBASE-M), Version 12.3(8)T3, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Tue 20-Jul-04 17:03 by eaarmas:
```

56

## Polling, utilisation de Net-SNMP

- Installer un agent Net-SNMP (chap. 6 livre SNMP 2nd ed. Mauro & Schmidt)
- Programme de configuration: *snmp.conf* (dans */usr/local/share/snmp*)

```
$ snmpget -v1 -c public orrouter1 .1.3.6.1.2.1.1.6.0
SNMPv2-MIB::sysLocation.0 = STRING: Sebastopol CA
```

```
$ snmpget -v1 -c public orrouter1 sysLocation.0
SNMPv2-MIB::sysLocation.0 = STRING: Sebastopol CA
```

57



## Récupération de plusieurs valeurs de la MIB

- *snmpwalk* à la place de *snmpget*

```
#!/usr/local/bin/perl
#filename: /opt/local/perl_scripts/snmpwalk.pl
use SNMP_util;
$MIB1 = shift;
$HOST = shift;
($MIB1) && ($HOST) || die "Usage: $0 MIB_OID HOSTNAME";
(@values) = &snmpwalk("$HOST", "$MIB1");
if (@values) { print "Results :$MIB1: :@values:\n"; }
else { warn "No response from host :$HOST:\n"; }
```

Here's how to run the script:

```
$ /opt/local/perl_scripts/snmpwalk.pl .1.3.6.1.2.1.2.2.1.2 orarouter1
```

```
Results :.1.3.6.1.2.1.2.2.1.2: :1:Ethernet0 2:Serial0 3:Serial1:
```

58

## Résultats de *snmpwalk*

Cisco 7000 router:

```
Results :.1.3.6.1.2.1.2.2.1.2: :1:Ethernet0/0 2:Ethernet0/1 3:TokenRing1/0
4:TokenRing1/1 5:TokenRing1/2 6:TokenRing1/3 7:Serial2/0 8:Serial2/1
9:Serial2/2 10:Serial2/3 11:Serial2/4 12:Serial2/5 13:Serial2/6 14:Serial2/7
15:FastEthernet3/0 16:FastEthernet3/1 17:TokenRing4/0 18:TokenRing4/1:
```

Linux workstation:

```
Results :.1.3.6.1.2.1.2.2.1.2: :1:lo 2:eth0 3:sit0:
```

Sun workstation:

```
Results :.1.3.6.1.2.1.2.2.1.2: :1:lo0 2:hme0:
```

Windows XP Pro PC:

```
Results :.1.3.6.1.2.1.2.2.1.2: :1:MS TCP Loopback interface
2: NETGEAR WG511 54 Mbps Wireless PC Card:
```

APC uninterruptible power supply:

```
Results :.1.3.6.1.2.1.2.2.1.2: :1:pda:
```

59

## Promenade dans l'arbre avec Net-SNMP

```
$ snmpwalk -v1 -c public orarouter1 .1.3.6.1.2.1.1
SNMPv2-MIB::system.sysDescr.0 = STRING: Cisco IOS Software, C2600 Software (C2600-
IPBASE-M), Version 12.3(8)T3, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2004 by Cisco Systems, Inc.
Compiled Tue 20-Jul-04 17:03 by eaarmas
SNMPv2-MIB::system.sysObjectID.0 = OID: enterprises.9.1.284
SNMPv2-MIB::system.sysUpTime.0 = Timeticks: (100946413) 11 days, 16:24:24.13
SNMPv2-MIB::system.sysContact.0 = STRING: thenetworkadministrator@oreilly.com
SNMPv2-MIB::system.sysName.0 = STRING: orarouter1@oreilly.com
SNMPv2-MIB::system.sysLocation.0 = STRING: Sebastopol CA
SNMPv2-MIB::system.sysServices.0 = STRING: 6
SNMPv2-MIB::system.sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

60

## Changement d'une valeur dans la MIB (*snmpset*)

```
#!/usr/local/bin/perl
#filename: /opt/local/perl_scripts/snmpset.pl
use SNMP_util;
$MIB1 = ".1.3.6.1.2.1.1.6.0";
$HOST = "oraswitch2";
$LOC = "@ARGV";
($value) = &snmpset("private@$HOST", "$MIB1", 'string', "$LOC");
if ($value) { print "Results :$MIB1: :$value:\n"; }
else { warn "No response from host :$HOST:\n"; }
```

Let's run this script:

```
$ /opt/local/perl_scripts/snmpset.pl A bld JM-10119 floor 7
Results :.1.3.6.1.2.1.1.6.0: :A bld JM-10119 floor 7:
```

Using the *snmpget.pl* script, we can verify that the set took place:

```
$ /opt/local/perl_scripts/snmpget.pl .1.3.6.1.2.1.1.6.0 public@oraswitch2
Results :.1.3.6.1.2.1.1.6.0: :A bld JM-10119 floor 7:
```

Now we'll use the Net-SNMP *snmpset* utility to change the system contact:

```
$ snmpset -v1 -c private oraswitch2 sysContact.0 s myself
SNMPv2-MIB::system.sysContact.0 = myself
$ snmpget -v1 -c public oraswitch2 sysContact.0
SNMPv2-MIB::system.sysContact.0 = myself
```

| Abbreviation | Meaning        |
|--------------|----------------|
| a            | IP address     |
| b            | Bits           |
| d            | Decimal string |
| D            | Double         |
| F            | Float          |
| i            | Integer        |

| Abbreviation | Meaning            |
|--------------|--------------------|
| I            | Signed int64       |
| O            | Object ID          |
| s            | String             |
| t            | Time ticks         |
| u            | Unsigned integer   |
| U            | Unsigned int64     |
| x            | Hexadecimal string |

61

## Erreurs

| Server responded with              | Explanation                                                                                                                                                                                                                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Contained under subtree            | snmpwalk returns this error if you have tried going down a MIB tree and are already at the end, or if the tree doesn't exist on the client.                                                                                                                                                   |
| No response arrived before timeout | Possible causes include an invalid community name, an agent that is not running, or an inaccessible node.                                                                                                                                                                                     |
| Agent reported error with variable | You are trying to set to an object with a datatype that is not the same as (or close to) the variable's specified type. For example, if the variable wants a DisplayString, you'll get this error if you send it an INTEGER. Read through the MIB to see what SYNTAX type the variable needs. |
| Missing instance value for...      | When you are setting a value, you must supply the entire OID and instance. A scalar object will end with zero (0) and a tabular object will end with the instance number of the object in a table. Verify that the instance number you're using with snmpget is correct and retry your set.   |
| Access is denied for variable      | This may happen if you are trying to set a value on a read-only object. Review the MIB to see what the object's ACCESS setting is.                                                                                                                                                            |

62

## Mis en place de niveaux d'alerte

- Actions à partir d'un certain niveau d'alerte.
- On peut faire un pooling externe (avec NMS) ou interne. Analogie: contrôle de l'huile dans la voiture (-> quel niveau? quelle fréquence (chaque mois, chaque semaine)?). L'huile qui manque peut être dû à un problème dans le moteur. Dans les réseaux, problème matériel? Trafic?
- Polling externe: utilise de la largeur de bande précieuse! -> essayer de faire du pooling interne (ne va pas utiliser NMS, ni l'agent SNMP), écrit en langage script en général (qui envoient des trapes à NMS s'il y a un problème)

63

## Exemple script pour le pooling interne

```
#!/usr/local/bin/perl
Filename: /opt/local/perl_scripts/check4file.pl

use SNMP_util "0.54"; # This will load the BER and SNMP_Session modules for us

$FILENAME = "/etc/passwd";

#
if the /etc/passwd file does not exist, send a trap!
#
if(!(-e $FILENAME)) {
 snmptrap("public@nms:162", ".1.3.6.1.4.1.2789", "sunserver1", 6, 1547, \
 ".1.3.6.1.4.1.2789.1547.1", "string", "File \:$FILENAME\: Could\
 NOT Be Found");
}
}
```

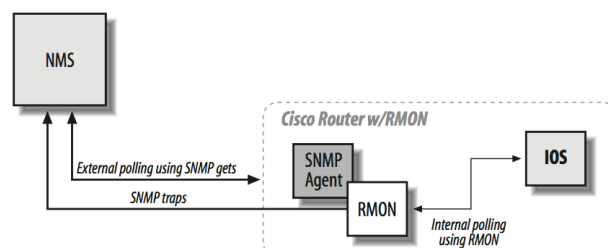
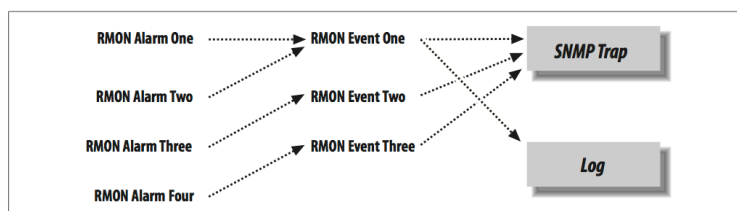
Here is what the Sun-style *crontab* looks like:

```
$ crontab -l

Check for this file every 15 minutes and report trap if not found
4,19,34,49 * * * * /opt/local/perl_scripts/check4file.pl
```

64

## Remote Monitoring (RMON)



65

## Configuration de RMON

```
xmon event number [log] [trap community] [description string] [owner string]
```

### *number*

Specifies the unique identification number for the event. This value must be greater than 0; a value of 0 is not allowed.

### log

Tells the agent to log the entry when triggered. This argument is optional.

### trap *community*

Specifies the trap community string, i.e., a community string to be included with the trap. Many network management programs can be configured to respond only to traps with a particular community string.

### description *string*

Describes the event.

### owner *string*

Ties the event or item to a particular person.

66

## Configuration des événements

```
(config)#xmon event 1 log trap public description "High ifInOctets" owner dmauro
(config)#xmon event 2 log description "Low ifInOctets" owner dmauro
```

```
orarouter1# show xmon event
```

```
Event 1 is active, owned by dmauro
Description is High ifInOctets
Event firing causes log and trap to community public, last fired 00:00:00
Event 2 is active, owned by dmauro
Description is Low ifInOctets
Event firing causes log, last fired 00:00:00
```

```
$ snmpwalk -v1 -c public -m orarouter1 .iso.org.dod.internet.mgmt.mib-2.xmon
RMON-MIB::eventIndex.1 = INTEGER: 1
RMON-MIB::eventIndex.2 = INTEGER: 2
RMON-MIB::eventDescription.1 = STRING: High ifInOctets
RMON-MIB::eventDescription.2 = STRING: Low ifInOctets
RMON-MIB::eventType.1 = INTEGER: logandtrap(4)
RMON-MIB::eventType.2 = INTEGER: log(2)
RMON-MIB::eventCommunity.1 = STRING: "public"
RMON-MIB::eventCommunity.2 = ""
RMON-MIB::eventLastTimeSent.1 = Timeticks: (0) 0:00:00.00
RMON-MIB::eventLastTimeSent.2 = Timeticks: (0) 0:00:00.00
RMON-MIB::eventOwner.1 = STRING: "dmauro"
RMON-MIB::eventOwner.2 = STRING: "dmauro"
RMON-MIB::eventStatus.1 = INTEGER: valid(1)
RMON-MIB::eventStatus.2 = INTEGER: valid(1)
```

67

## Configuration d'alarmes en pooling interne

- On veut surveiller un interface du routeur (serial interface):  
`mngr.interfaces.ifEntry.  
ifInOctets.2` (nombre d'octets qui sortent de l'interface). Alerte si le trafic est supérieur à 90 kb/s et lorsqu'il redevient inférieur à 85 kb/s.  
Pooling chaque 60 sec.

```
xmon alarm number variable interval {delta | absolute}

 rising-threshold value [event-number]

 falling-threshold value [event-number]

 [owner string]
```

### *number*

Specifies the unique identification number assigned to the alarm.

### *variable*

Specifies which MIB object to monitor.

### *interval*

Specifies the frequency (in seconds) at which the alarm monitors the MIB variable.

### *delta*

Indicates that the threshold values given in the command should be interpreted in terms of the difference between successive readings.

### *absolute*

Indicates that the threshold values given in the command should be interpreted as absolute values; i.e., the difference between the current value and preceding values is irrelevant.

### *rising-threshold value [event-number]*

Specifies the *value* at which the alarm should be triggered, calling the event, when the value is rising. *event-number* is the event that should be called when the alarm occurs. The event number is optional because the threshold doesn't have to be assigned an event. If either of the two thresholds is left blank, the event number will be set to 0, which does nothing.

### *falling-threshold value [event-number]*

Specifies the *value* at which the alarm should be triggered, calling the event, when the value is falling. *event-number* is the event that should be called when the alarm occurs. The event number is optional because the threshold doesn't have to be assigned an event. If either of the two thresholds is left blank, the event number will be set to 0, which does nothing.

### *owner string*

Ties this alarm to a particular person.

68

## Configuration des alarmes

```
orrouter1(config)#xmon alarm 25 ifEntry.10.2 60 absolute \

 rising-threshold 90000 1 falling-threshold 85000 2 owner dmauro
```

```
orrouter1#show xmon alarm
```

```
Alarm 1 is active, owned by dmauro

Monitors ifEntry.10.2 every 60 second(s)

Taking absolute samples, last value was 87051

Rising threshold is 90000, assigned to event 1

Falling threshold is 85000, assigned to event 2

On startup enable rising or falling alarm
```

- Parfois le pooling interne est difficile à grapher. Les tendances sont parfois pénibles à obtenir (MRTG le fait).

69

## Trapes

- Qu'est-ce qu'une trape? Notification envoyée par un agent SNMP à un station de management, en utilisant UDP.

Table 2-8. Generic traps

| Generic trap name and number | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coldStart (0)                | Indicates that the agent has rebooted. All management variables will be reset; specifically, Counters and Gauges will be reset to zero (0). One nice thing about the coldStart trap is that it can be used to determine when new hardware is added to the network. When a device is powered on, it sends this trap to its trap destination. If the trap destination is set correctly (i.e., to the IP address of your NMS), the NMS can receive the trap and determine whether it needs to manage the device. |
| warmStart (1)                | Indicates that the agent has reinitialized itself. None of the management variables will be reset.                                                                                                                                                                                                                                                                                                                                                                                                            |
| linkDown (2)                 | Sent when an interface on a device goes down. The first variable binding identifies the index in the <i>interfaces</i> table for the interface that went down.                                                                                                                                                                                                                                                                                                                                                |
| Generic trap name and number | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| linkUp (3)                   | Sent when an interface on a device comes back up. The first variable binding identifies which interface came back up.                                                                                                                                                                                                                                                                                                                                                                                         |
| authenticationFailure (4)    | Indicates that someone has tried to query your agent with an incorrect community string; useful in determining if someone is trying to gain unauthorized access to one of your devices.                                                                                                                                                                                                                                                                                                                       |
| egpNeighborLoss (5)          | Indicates that an EGP neighbor has gone down.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| enterpriseSpecific (6)       | Indicates that the trap is enterprise-specific. SNMP vendors and users define their own traps under the <i>private-enterprise</i> branch of the SMI object tree. To process this trap properly, the NMS has to decode the specific trap number that is part of the SNMP message.                                                                                                                                                                                                                              |

70

## Trapes

- Manière pour un agent d'envoyer des notifications de manière asynchrone sur les conditions du réseau.
- Les trapes qu'un agent peut générer sont définies dans la MIB.
- Configuration du NMS en réponse aux différentes trapes (ignorer, script,...)
- Openview ou Perl (solution plus abordable)

71

## Trapes (2)

| Generic trap name and number | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| coldStart (0)                | Indicates that the agent has rebooted. All management variables will be reset; specifically, Counters and Gauges will be reset to zero (0). One nice thing about the coldStart trap is that it can be used to determine when new hardware is added to the network. When a device is powered on, it sends this trap to its trap destination. If the trap destination is set correctly (i.e., to the IP address of your NMS), the NMS can receive the trap and determine whether it needs to manage the device. |
| warmStart (1)                | Indicates that the agent has reinitialized itself. None of the management variables will be reset.                                                                                                                                                                                                                                                                                                                                                                                                            |
| linkDown (2)                 | Sent when an interface on a device goes down. The first variable binding identifies the index in the <i>interfaces</i> table for the interface that went down.                                                                                                                                                                                                                                                                                                                                                |
| Generic trap name and number | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| linkUp (3)                   | Sent when an interface on a device comes back up. The first variable binding identifies which interface came back up.                                                                                                                                                                                                                                                                                                                                                                                         |
| authenticationFailure (4)    | Indicates that someone has tried to query your agent with an incorrect community string; useful in determining if someone is trying to gain unauthorized access to one of your devices.                                                                                                                                                                                                                                                                                                                       |
| egpNeighborLoss (5)          | Indicates that an EGP neighbor has gone down.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| enterpriseSpecific (6)       | Indicates that the trap is enterprise-specific. SNMP vendors and users define their own traps under the <i>private-enterprise</i> branch of the SMI object tree. To process this trap properly, the NMS has to decode the specific trap number that is part of the SNMP message.                                                                                                                                                                                                                              |

- Port UDP 162
- Définis de 0 à 6 (coldStart: reboot, linkUp/linkDown: chgt état interface, enterpriseSpecific: trape générique)

72

## Recevoir des trapes (Perl)

```
#!/usr/bin/perl
```

Programme de Simon Leinen

```
use SNMP_Session;
use BER;
use Socket;

$session = SNMPv1_Session->open_trap_session ();
while (($trap, $sender, $sender_port) = $session->receive_trap ()) {
 chomp ($DATE = `bin/date +%a %b %e %T`);
 print STDERR "\n$DATE - " . inet_ntoa($sender)
 . " - port: $sender_port\n";
 print_trap ($session, $trap);
}

sub print_trap {
 ($this, $trap) = @_;
 my($community, $sent, $agent, $gen, $spec, $dt, $bindings)
 = $this->decode_trap_request ($trap);
 print " Community:\t". $community. "\n";
 print " Enterprise:\t". BER::pretty_oid ($sent). "\n";
 print " Agent addr:\t". inet_ntoa ($agent). "\n";
 print " Generic ID:\t$gen\n";
 print " Specific ID:\t$spec\n";
 print " Uptime:\t". BER::pretty_uptime_value ($dt). "\n";
 $prefix = " bindings:\t";
 my ($binding, $oid, $value);
 while ($bindings ne '') {
 ($binding, $bindings) = &decode_sequence ($bindings);
 ($oid, $value) = decode_by_template ($binding, "%0%@");
 print $prefix. BER::pretty_oid ($oid).
 " => ". pretty_print ($value). "\n";
 $prefix = "\t\t";
 }
}
```

Sortie

```
Mon Apr 28 22:07:44 - 10.123.46.26 - port: 63968
community: public
enterprise: 1.3.6.1.4.1.2789.2500
agent addr: 10.123.46.26
generic ID: 6
specific ID: 5247
uptime: 0:00:00
bindings: 1.3.6.1.4.1.2789.2500.1234 => 14264026886

Mon Apr 28 22:09:46 - 172.16.51.25 - port: 63970
community: public
enterprise: 1.3.6.1.4.1.2789.2500
agent addr: 172.16.253.2
generic ID: 6
specific ID: 5247
uptime: 0:00:00
bindings: 1.3.6.1.4.1.2789.2500.2468 => Hot Swap Now In Sync
```

73



## Recevoir des trapes (Net-SNMP)

- Net-SNMP est gratuit!
- Commande ci-dessous: transmet des trapes en format standard (-Lo) et pas syslog, pas de « fork » (-f)

```
$./snmptrapd -f -Lo
2005-05-05 08:00:24 NET-SNMP version 5.2.1 Started.
2005-05-05 08:03:05 sunserver2.ora.com [12.1.45.26] (via UDP: [12.1.45.26]:37223)
TRAP, SNMP v1, community public
 SNMPv2-SMI::enterprises.2789.2500.1224 Enterprise Specific Trap (1224)
Uptime: 60 days, 14:41:38.72
 SNMPv2-SMI::enterprises.2789.2500.1224 = INTEGER: 123123

2005-05-05 08:10:16 sunserver2.ora.com [12.1.45.26] (via UDP: [12.1.45.26]:37223)
TRAP, SNMP v1, community public
 SNMPv2-SMI::enterprises.2789.2500.1445 Enterprise Specific Trap (1445) Uptime:
60 days, 14:41:38.72
 SNMPv2-SMI::enterprises.2789.2500.1445 = STRING: "Fail Over Complete"
```

74

## Envoyer des trapes

- On peut utiliser des programmes ou des scripts déjà existants
- Arguments:

Port, SNMP version, hostname or IP address du NMS, community name, enterprise OID, hostname or IP address sender, generic trap number, timestamp

75

## Envoyer des trapes (Perl)

---

- appel de snmptrap

```
snmptrap(communityname@host:port_number, enterpriseOID, host_name_from, \
generic_ID, specific_ID, OID, type, value, [OID, type, value ...])
```

- Génération d'une trape avec une valeur

```
#!/usr/local/bin/perl
Filename: /opt/local/perl_scripts/snmptrap.pl

use SNMP_util "0.54"; # This will load the BER and SNMP_Session for us

snmptrap("public@nms:162", ".1.3.6.1.4.1.2789", "sunserver1", 6, 1247, \
".1.3.6.1.4.1.2789.1247.1", "int", "2448816");
```

- Avec plusieurs valeurs

```
snmptrap("public@nms:162", ".1.3.6.1.4.1.2789", "sunserver2", 6, 3301, \
".1.3.6.1.4.1.2789.3301.1", "int", "4278475", \
".1.3.6.1.4.1.2789.3301.2", "string", "Sybase DB Stopped");
```

76

## Envoyer des trapes (Perl)

---

- Pour le monitoring des trapes générées (avec Net-SNMP):

```
$./snmptrapd -f -Lo
2005-05-05 08:00:24 NET-SNMP version 5.2.1 Started.
2005-05-05 08:03:05 sunserver2.ora.com [12.1.45.26] (via UDP: [12.1.45.26]:37223)
TRAP, SNMP v1, community public
 SNMPv2-SMI::enterprises.2789.3301 Enterprise Specific Trap (3301) Uptime: 60
days, 14:41:38.72
 SNMPv2-SMI::enterprises.2789.3301.1 = INTEGER: 4278475
 SNMPv2-SMI::enterprises.2789.3301.2 = STRING: "Sybase DB Stopped"
```

77

## Envoyer des trapes (Net-SNMP)

---

- Comment le program est appelé:

```
snmptrap hostname community enterprise-oid agent \
generic-trap specific-trap uptime [OID type value]...
```

- Génération d'une trape avec une seule valeur:

```
$ snmptrap nms public .1.3.6.1.4.1.2789.2005 ntserver1 6 2476317 '' \
.1.3.6.1.4.1.2789.2005.1 s "WWW Server Has Been Restarted"
```

- SNMPv2:

```
$ snmptrap -v2c nms public '' .1.3.6.1.6.3.1.1.5.3 \
ifIndex i 2 ifAdminStatus i 1 ifOperStatus i 1
```

- SNMPv3

```
$ snmptrap -e 0x012345 -v3 -l authPriv -u kschnidt -a MD5 \ -A mysecretpass -x DES -X
mypassphrase localhost '' \ .1.3.6.1.4.1.2789.2005 .1.3.6.1.4.1.2789.2005.1 s \
"WWW Server Has Been Restarted"
```

78

## Service Monitoring

---

79

## Outils pour le dépannage (rappel)

---

- Ping
- ipconfig-ifconfig
- arp
- netstat
- traceroute/tracert
- whois
- Wireshark