

Le protocole SNMP

SNMP

1

Les composants d'un gestionnaire réseau OSI

- Gestion des erreurs (détection, isolement correction);
- Utilisation (affecter un coût d'utilisation à une utilisation);
- Configuration et nommage (détermination des données à récupérer);
- Performance (analyse le comportement des objets gérés et l'efficacité du réseau);
- Sécurité (sécurité au niveau de l'outil de gestion réseau).

SNMP

3

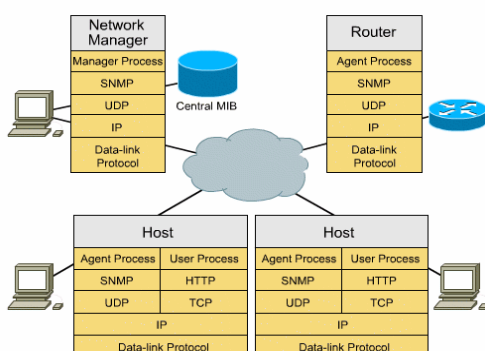
Les protocoles d'administration réseau

- **CMOT (CMIP Over TCP)**
 - le protocole CMIP (Common Management Information Protocol) utilise la norme ISO, il est très lourd à mettre en oeuvre. CMOT est une adaptation du protocole pour fonctionner au dessus de la couche transport de l'Internet : TCP). CMIP est peu utilisé actuellement.

SNMP

5

Architecture SNMP



SNMP

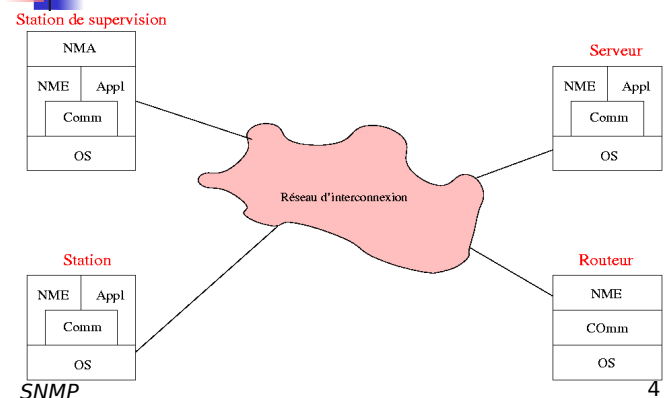
7

- Quand un réseau de ressources informatiques devient trop important il ne peut plus être géré efficacement par un homme sans outil automatisé.
- Les fonctions d'un gestionnaire de réseau
 - Contrôle d'une stratégie d'entreprise;
 - Contrôle de la complexité;
 - Uniformité des services;
 - Contrôle de la performance, de la sécurité et de l'accessibilité en fonction des utilisateurs et des besoins spécifiques;
 - Réduction maximum des temps d'inaccessibilité;
 - Contrôle du coût d'utilisation.

SNMP

2

Architecture d'une gestion réseau



SNMP

4

Les protocoles d'administration réseau

- **SNMP (Simple Network Management Protocol)**
 - Le protocole SNMP est très utilisé, il est basé sur le protocole de l'internet (TCP/IP)
 - Les éléments du protocole:
 - Les stations administrées (les agents);
 - La station d'administration ou de supervision (le client)
 - Le protocole d'administration réseau qui permet les échanges entre les agents et le (ou les) client.
 - **get** : le client récupère une donnée de l'agent;
 - **getNext** : le client récupère la donnée suivante sur l'agent;
 - **set** : le client modifie une donnée sur l'agent;
 - **trap** : l'agent envoie un événement vers le client.

SNMP

6

Organisation des informations d'administration

- Une **MIB (Management Information Base)** contient les informations représentant l'état d'un agent. Pour que la MIB soit utilisable par tous les agents et les processus de supervision, elle doit suivre les règles suivantes:
 - Un objet qui représente une ressource particulière doit être le même quelque soit l'agent;
 - Un schéma commun pour la représentation des données doit être utilisé pour une architecture hétérogène.

SNMP

8

Organisation des informations d'administration

- Les règles suivent la norme **SMI** (*Structure of Management Information*) qui est décrite par la notation **ASN.1** (*Abstract Syntax Notation One*).

La SMI propose:

- Une technique de standardisation pour définir la structure d'une MIB;
- Une technique de standardisation pour définir un objet avec sa syntaxe et son domaine de valeur;
- Une technique de standardisation pour coder les valeurs d'un objet.

SNMP

9

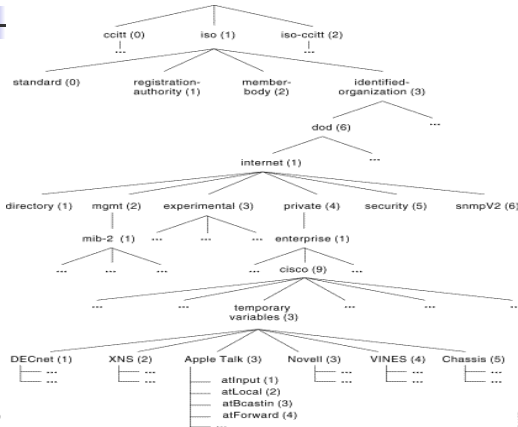
La structure d'une MIB

- Chaque objet manipulé par SNMP appartient à une structure d'arbre.
- Chaque objet d'une MIB est identifié par un **identificateur** (**OID** : *Object Identifier*).
- Un OID est une suite de nombres séparés par des points.
- Les feuilles de l'arbre contiennent les données accessibles de l'agent.
- Il y a deux types de structure de feuilles :
 - Scalaire
 - Tableau
- Une MIB contient la description des données, pas les valeurs

SNMP

10

La structure d'une MIB



SNMP

11

Un objet scalaire

- Un **objet scalaire** est un objet qui n'est pas contenu dans un tableau.
- Pour accéder à une instance (valeur concrète) d'un objet scalaire, le client ajoute '.0' à son identificateur
- Par exemple la donnée qui représente le nombre d'erreurs pendant les connexions TCP est définie dans une MIB standard par l'identificateur suivant : **.1.3.6.1.2.6.14**.
- Si un client (station d'administration) veut obtenir la valeur de l'instance de cette donnée, il faut indiquer dans le message la référence suivante : **.1.3.6.1.2.6.14.0**.

SNMP

12

Un objet tableau

- Un **objet tableau** est un objet qui peut contenir plusieurs instances.

SEQUENCE OF <entry>

- Les objets (lignes d'une table ou *entry*) contenus dans le tableau peuvent être des séquences d'éléments (plusieurs éléments simples).
- Chaque séquence (ligne) du tableau doit contenir les mêmes éléments.

SEQUENCE {<type1> ... <typeN>}

- La taille de ce tableau n'est pas bornée.

SNMP

13

Un objet tableau

- Pour accéder à une instance particulière du tableau, on ajoute à l'identificateur de l'objet une valeur d'index (qui peut être plus ou moins complexe). Par exemple l'objet **ifDesc** **.1.3.6.1.2.1.2.2.1.2** représente la deuxième colonne de la ligne **fEntry** **(.1.3.6.1.2.1.2.2.1)** dans le tableau **ifTable** **(.1.3.6.1.2.1.2.2)**, pour y accéder on utilisera l'identificateur **.1.3.6.1.2.1.2.2.1.2.i** avec **i** la valeur de l'index

SNMP

14

Définition d'un objet d'une MIB

- Chaque objet dans une MIB suit la structure définie dans la syntaxe ASN.1

```
OBJECT-TYPE MACRO ::=
BEGIN
  TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
    "ACCESS" Access
    "STATUS" Status
    DescrPart
    ReferPart
    IndexPart
    DefValPart
  VALUE NOTATION ::= value (VALUE ObjectName)

  Access ::= "read-only" | "read-write" | "write-only" | "not-accessible"
  Status ::= "mandatory" | "optional" | "obsolete" | "deprecated"
  DescrPart ::= "DESCRIPTION" value (description DisplayString) | empty
  ReferPart ::= "REFERENCE" value (reference DisplayString) | empty
  IndexPart ::= "INDEX" "{" IndexTypes "}"
  IndexTypes ::= IndexType | IndexTypes "," IndexType
  IndexType ::= value (indexobject ObjectName) | type(indextype)
  DefValPart ::= "DEFVAL" "{" value (defvalue ObjectSyntax) "}" | empty
  DisplayString ::= OCTET STRING SIZE (0..255)
END
```

SNMP

15

Exemple d'un objet scalaire

- L'objet **tcpMaxConn**

```
tcpMaxConn OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The limit on the total number of TCP connections
     the entity can support. In entities where the
     maximum number of connections is dynamic, this
     object should contain the value -1."
  ::= {tcp 4}
```

SNMP

16

Un exemple de table

Le tableau tcpConnTable

```
tcpConnTable OBJECT-TYPE
SYNTAX SEQUENCE OF TcpConnEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A table containing TCP connection-specific information"
::= { tcp 13 }
tcpConnEntry OBJECT-TYPE
SYNTAX TcpConnEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "..."
INDEX { tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort }
::= { tcpConnTable 1 }
TcpConnEntry ::= SEQUENCE {
    tcpConnState INTEGER,
    tcpConnLocalAddress IPADDRESS,
    tcpConnLocalPort INTEGER (0..65535),
    tcpConnRemAddress IPADDRESS,
    tcpConnRemPort INTEGER (0..65535)
}
```

SNMP

17

Les types des Objets d'une MIB

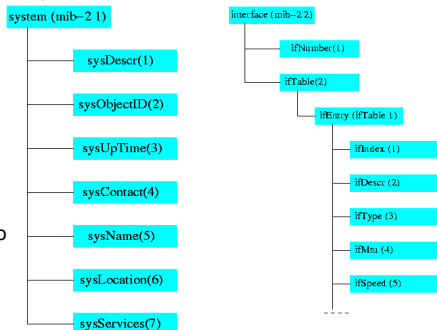
- Les types simples
 - INTERGER
 - OCTET STRING
 - OBJECT IDENTIFIER
 - NULL
 - SEQUENCE et SEQUENCE OF
- Les types applicatifs
 - NetworkAddress (CHOICE contenant uniquement IpAddress)
 - IpAddress
 - Counter (Counter32 ou Counter64 en SNMP2) repasse à 0 lorsque = max maximum ($2^{32} - 1$)
 - Gauge ne repasse pas à 0, maximum ($2^{32} - 1$)
 - TimeTicks compteur en 1/100 seconde depuis une origine
 - Opaque représente un codage arbitraire

SNMP

18

La MIB standard (mib-2)

- La MIB **mib-2** est le premier fils de **mgmt** dans l'arbre de nommage des objets. Cette MIB est constituée de 10 groupes:

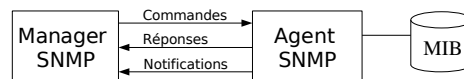


SNMP

19

Le protocole SNMP

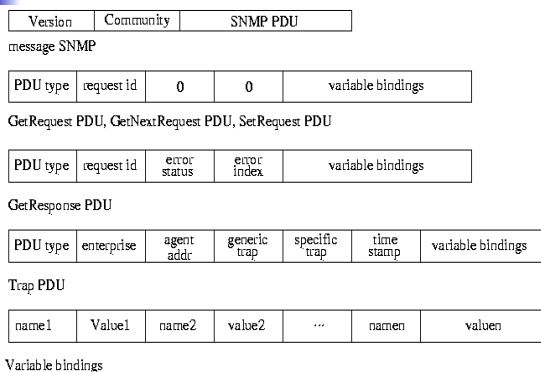
- Le protocole SNMP fait de la couche application dans le réseau Internet
- Il utilise le protocole UDP de la couche transport et
- Il est affecté aux ports 161 et 162 du protocole UDP par défaut



SNMP

20

Le protocole SNMP

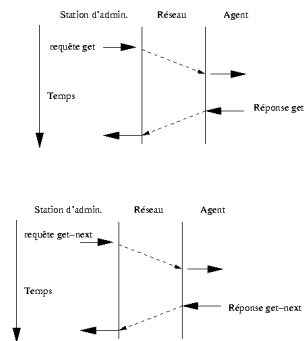


SNMP

21

Les opérations du protocole SNMP

- Get : La station de supervision reçoit un scalaire de la station administrée
- Get Next : La station de supervision reçoit le scalaire de la station administrée qui suit (lexicographiquement) le scalaire de la requête précédente.

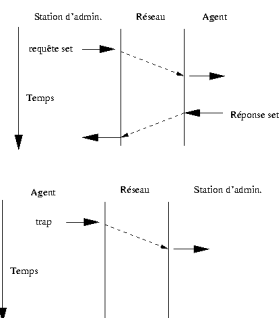


SNMP

22

Les opérations du protocole SNMP

- Set : La station de supervision modifie un scalaire de la station administrée

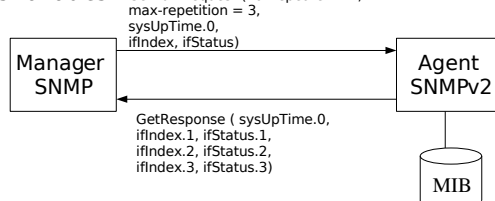


SNMP

23

Les opérations du protocole SNMP

- Avec SNMPv2 on trouve une autre primitive GetBulk qui permet dans la même requête de mélanger la primitive Get et GetNext.
- La requête indique le nombre de variables à traiter comme une requête Get et le nombre de requêtes GetNext pour les autres variables



SNMP

24

Les droits d'accès

■ L'authentification

- Dans chaque message échangé entre la station de supervision et les agents on trouve un **nom** de groupe d'utilisateurs (**community name**).
- Le **nom** joue le rôle d'un mot de passe, c'est à dire que l'agent considère que si la station de supervision connaît le nom du groupe alors elle est authentifiée et autorisée à agir comme un membre du groupe.

SNMP

25

Les méthodes d'accès aux données gérées par un agent

■ L'accès direct : exemple pour une table

TcpConnState (1.3.6.1.2.1.6.13.1.1)	TcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2)	TcpConnLocalPort (1.3.6.1.2.1.6.13.1.3)	TcpConnRemAddress (1.3.6.1.2.1.6.13.1.4)	TcpConnRemPort (1.3.6.1.2.1.6.13.1.5)	
5	10.0.0.99	12	9.1.2.3	15	TcpConnEntry(1.3.6.1.2.1.6.13.1)
2	10.0.0.99	99	192.168.37.5	25	TcpConnEntry(1.3.6.1.2.1.6.13.1)
3	10.0.0.99	14	89.1.1.42	84	TcpConnEntry(1.3.6.1.2.1.6.13.1)

TcpConnState (1.3.6.1.2.1.6.13.1.1)	TcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2)	TcpConnLocalPort (1.3.6.1.2.1.6.13.1.3)	TcpConnRemAddress (1.3.6.1.2.1.6.13.1.4)	TcpConnRemPort (1.3.6.1.2.1.6.13.1.5)
x.1.10.0.0.99.12.9.1.2.3 15	x.2.10.0.0.99.12.9.1.2.3 15	x.3.10.0.0.99.12.9.1.2.3 15	x.4.10.0.0.99.12.9.1.2.3 15	x.5.10.0.0.99.12.9.1.2.3 15
x.1.10.0.0.99.14.89.1.1 42.84	x.2.10.0.0.99.14.89.1.1 2.84	x.3.10.0.0.99.14.89.1.1 42.84	x.4.10.0.0.99.14.89.1.1 2.84	x.5.10.0.0.99.14.89.1.1 42.84

SNMP

27

Les méthodes d'accès aux données gérées par un agent

■ L'accès série

- L'identificateur d'un objet porte dans son écriture (suite d'entiers) une représentation hiérarchique de la structure qui la contient.
- La suite d'entiers permet d'utiliser un ordre lexicographique
- Les noeuds fils sont définis en ajoutant un entier à la liste des entiers de l'identificateur du père et en visitant l'arbre de bas en haut et de la gauche vers la droite.
- Cet accès série permet d'accéder à des objets sans en connaître l'identificateur exact.

SNMP

29

Echange sur le réseau pour le protocole SNMP

■ Emission d'un message

- Construction du PDU qui va contenir la requête
 - Type de message (get, get-next, set, réponse ou trap)
 - Génération d'un numéro d'identification de la requête
 - Liste des couples (variables ,valeur) que l'on veut échanger
- Ajout d'un nom de communauté de la version

■ Réception d'un message

- Analyse du message
 - Examen du nom de la communauté
 - Examen du PDU (numéro de requête, les couples (variables, valeur))

SNMP

31

Les droits d'accès

■ La politique d'accès

- L'agent peut limiter l'accès à une sélection de stations d'administration
- L'agent peut définir plusieurs groupes (**community name**).
- Le contrôle d'accès se compose de deux aspects
 - Une **vue** de la MIB: un sous ensemble de la MIB (plusieurs vues sont possibles pour chaque communauté)
 - Un **mode d'accès** {**read-only**, **read-write**}
- La vue et le mode d'accès forment le profil de la communauté SNMP.

SNMP

26

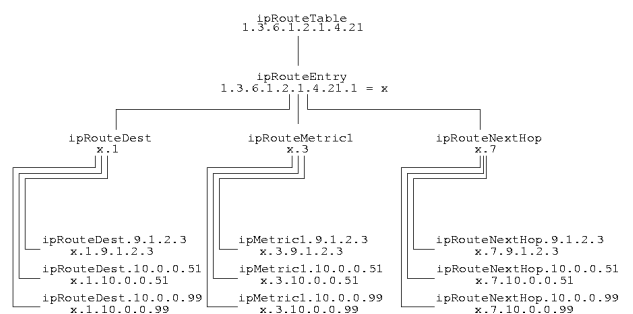
Les méthodes d'accès aux données gérées par un agent

- Sélection de la colonne dans la table : on utilise l'identificateur (OID)
- Sélection de la ligne dans la table : on utilise l'index défini pour la table
 - Soit y la valeur de l'OID de la donnée que l'on veut accéder
 - Soit i1, i2, ...iN les objets qui constituent l'index
 - Alors pour accéder à une colonne et une ligne particulière dans la table on utilisera l'identifiant suivant :
 - y.i1.i2.....iN
- Par exemple si l'on veut obtenir l'état de la connexion TCP entre les machines 10.0.0.99 sur le port 14 et 89.1.1.42 sur le port 84, la valeur de l'identifiant sera :

SNMP

28

Les méthodes d'accès aux données gérées par un agent



SNMP

30

Echange sur le réseau pour le protocole SNMP

■ Get

- Réception d'une **GetRequest-PDU**,
- Si pour chaque objet de la **VarBindList**, l'objet ne correspond pas alors envoi d'un **GetResponse-PDU** avec : **ErrorStatus**<-- NoSuchName et **ErrorIndex**<-- Valeur fausse dans le message reçu,
- Si **GetResponse-PDU** > Limitation locale alors envoi de **GetResponse-PDU** avec **ErrorStatus**<--too big et **ErrorIndex**<--0
- Si une des variables demandées ne peut pas être obtenue alors envoi de **GetResponse-PDU** avec **ErrorStatus**<--generr et **ErrorIndex**<-- variable en erreur
- Si tout est OK, envoi d'un **GetResponse-PDU** où les variables sont associées aux valeurs

SNMP

32

Échange sur le réseau pour le protocole SNMP

Get-Next

- Réception d'un **GetNextRequest-PDU**
- Si, pour un objet de la **varBindList**, le nom ne précède pas (lexicographiquement) le nom d'un objet accessible par un get, alors un **GetResponse-PDU** est renvoyé avec le même contenu et : **ErrorStatus** <-- NoSuchName et **ErrorIndex** <-- pointe sur la variable non ok dans la demande
- Si **GetResponse-PDU** > limitation locale alors envoi de **GetResponse-PDU** avec : **ErrorStatus** <--too big et **ErrorIndex** <-- 0
- Si une des variables demandées ne peut pas être obtenue alors envoi de **GetResponse-PDU** avec : **ErrorStatus** <--generr et **ErrorIndex** <-- variable en erreur
- Si tout est OK, envoi d'un **GetResponse-PDU** où les variables sont associées à des valeurs

33

SNMP

Échange sur le réseau pour le protocole SNMP

Set

- Réception d'un message **SetRequest-PDU**
- Si, pour chaque objet du champ **Variable-Bindings**, l'objet n'est pas accessible pour l'opération demandée alors la PDU **GetResponse-PDU** (de forme identique) est envoyée avec **ErrorStatus** <-- NoSuchName et **ErrorIndex** <-- pointeur sur la variable en erreur
- Si, pour chaque objet de la **VarBindList**, la valeur ne correspond pas au type attendu (longueur, valeur,...) alors la PDU **GetResponse-PDU** (de forme identique) est envoyée avec **ErrorStatus** <--BadValue et **ErrorIndex** <-- variable en erreur
- Si **GetResponse-PDU** > limitation locale alors envoi de **GetResponse-PDU** avec **ErrorStatus** <--toobig et **ErrorIndex** <--0
- Si une des variables de la **VarBindList** ne peut pas être mise à jour alors **GetResponse-PDU** est envoyée avec **ErrorStatus** <--generr et **ErrorIndex** <-- variable en erreur
- Si tout est OK, les variables citées dans la **VarBindList** sont mises à jour et un **GetResponse-PDU** est envoyé (avec un contenu identique) avec : **Error_status** <-- 0 et **ErrorIndex** <--0

34

SNMP

Échange sur le réseau pour le protocole SNMP

- Trap : l'agent transmet vers la station de supervision un message avec un champ **generic-trap** qui précise la nature de l'erreur
 - **coldstart** : l'agent envoyant le trap se réinitialise suite à un incident (crash, erreur majeure, ...). Le redémarrage n'était pas prévu
 - **warmstart** : l'agent envoyant le trap se réinitialise suite à une altération de ses données
 - **linkdown** : signale l'erreur sur une voie de communication de l'agent. Le premier élément de la **VarBindList** précise l'interface en erreur
 - **linkup** : signale qu'une voie de communication de l'agent est mise en service. Le premier élément de la **VarBindList** précise l'interface activée

35

SNMP

Échange sur le réseau pour le protocole SNMP

Trap : (suite)

- **authenticationfailure** : signale que l'agent a reçu un message non authentifié
- **egpneihborloss** : le routeur voisin de l'agent qui communiquait avec lui via EGP vient d'être stoppé
- **enterprisespecific** : indique qu'un événement spécifique vient de se produire. Le specific trap indique le numéro de trap concerné.

36

SNMP