Administration du réseau

(/home/kouna/d01/adp/bcousin/Fute/Cours/Internet-2/14-SNMP.fm- 10 Octobre 1998 13:23)

PLAN

- Introduction
- Présentation générale
- Le protocole SNMP
- La base de donnés - MIB
- La représentation des données
- Les messages SNMP
- Conclusion

1. Introduction

Buts de l'administration de réseau :

. configuration (configuration management)
. sécurité (security management)
. panne (fault management)
. audit (performance management)
. comptabilité (accounting management)

Le réseau est hétérogène :

. un seul ensemble d’opérations et un protocole d'administration (SNMP)
. une seule base d’information répartie (MIB)

Indépendance vis-à-vis des applications et des interfaces

Le réseau est réparti :

. administration à distance
. s'appuie sur le réseau lui-même (IP + UDP)

2. Présentation

2.1. Gestion de la configuration

Principaux rôles :

- inventaire des ressources
- initialisation des équipements
- gestions des noms et des adresses
- mise à jour des paramètres des ressources

Procédures :

- collecter les informations
- contrôler l'état du système
- sauvegarder l'historique ("log")
- présenter l'état du système ☞ synoptique

2.2. Gestion de la sécurité

Nécessaire pour protéger le réseau contre :

- un dysfonctionnement, une inadvertance, une malveillance

Une base de données spécifique :

- Security MIB : rassemble (protège) les informations utilisées pour assurer la sécurité

Attaque passive :

- écoute de messages, observation du trafic

Attaque active :

- mascarade,
- duplication de message, modification de message,
- perturbation d'un service, modification d'un service.

Enregistrement de l'activité des utilisateurs :

- les évènements significatifs, les actions interdites ou sensibles

Filtrage ("firewall")

- trie des flux de données circulant entre deux parties du réseau (inter/intranet)

- sur les adresses IP, le sens d'entrée/sortie, le type du protocole, le numéro de port,
- les applications accessibles doivent être protégées

Authentification :

- de l'utilisateur de service (mot de passe), de l'émetteur de message (signature)
- notaire (tierce partie certifiante)

Intégrité du message :

- sceau (fonction de hachage)

Cryptage des messages :

- système à clef secrète (symétrique) ou publique (asymétrique)
- cryptage à la source ou par un serveur

2.3. Gestion des pannes

Défauts :

- systématique : panne d'un équipement, rupture d'un lien
- dépendant : fonction de l'état de l'environnement, congestion, etc.

Phase de traitement d'un défaut :

- détection d'un fonctionnement anormal
- localisation/diagnostic
- réparation
- vérification

Détection :

- messages d'erreur (quoi, qui(où), quand)
- tests (de contrôle routinier, de diagnostic)
- seuils (dénombrement des évènements)

Diagnostic :

- exploitation de l'historique (suite d'évènements, ensemble d'évènements)
- tests de diagnostic

Réparation :

- reconfiguration
- remplacement

2.4. Audit des performances

Performances des ressources du réseau :

- délai, débit, taux d’erreur, disponibilité

Evaluation des performances effectuée à partir de mesures statistiques :

- Collecte,
- Contrôle,
- Stockage,
- Présentation

Analyse :

- Détection de comportements symptomatiques
- Prévision

2.5. Gestion de la comptabilité

Informations permettant d'évaluer le coût des communications.

- En fonction de la durée, du volume
- Au niveau du réseau ou de l'application

Exemple :

- nombre d'octets transmis, durée de connexion

3. SNMP

3.1. Introduction

SNMP (Simple Network Management Protocol): rfc 1157 (1990)
- utilise UDP : transmission simple !

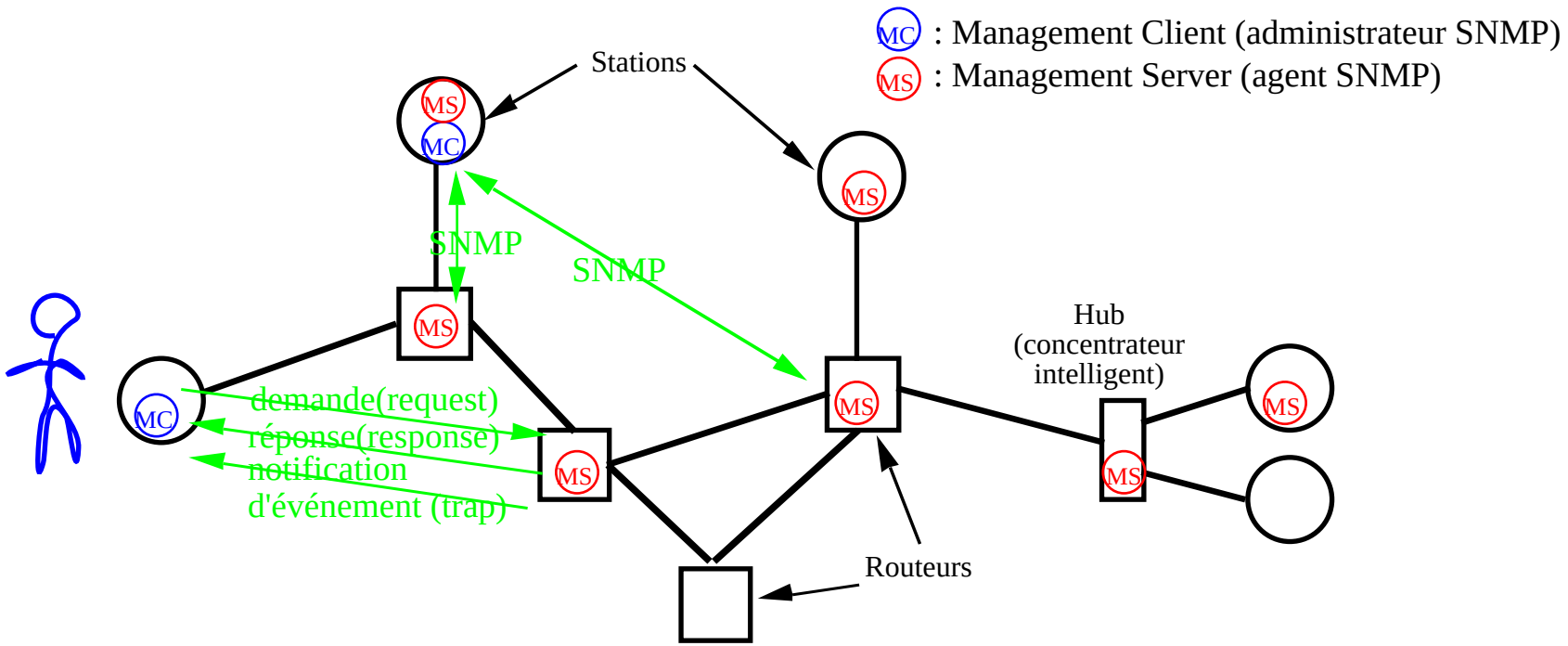
Norme OSI d'administration de réseau (ISO 7498) :
- CMIS/CMIP (ISO 9595 et 9596) :
  Common Management Information Service/Protocol
  ☞ CMOP (CMIP over TCP) : rfc1189.

MIB (Management Information Base) : rfc 1156
☞ Base de données répartie

⊕ . indépendance vis-à-vis des protocoles (uniforme)

⊖ . uniformité ≠ adaptabilité aux différents besoins
(Ethernet : rfc 1398, FDDI : rfc 1512, pont : rfc 1493, DEC : rfc 1289)
☞ MIB-II : rfc 1213 (1993)

3.2. Architecture générale



L'agent est chargé de gérer les équipements : il en propose une certaine vue.
L'administrateur peut interroger/piloter les équipements par l'intermédiaire des agents.
Des proxys peuvent être utilisés pour réaliser une adaptation (d'équipement ou protocolaire)

4. La base de données

4.1. Introduction

Management Information Base

Définit les informations gérées par la base de données :
Ces informations pourront être interrogées/modifiées par les administrateurs

Catégories d'information d'administration :
- system : système et informations générales
- interfaces : interface d'accès au réseau (coupleur, contrôleur, …)
- addr.trans. : adressage (ARP...)
- ip : protocole IP
- tcp : protocole TCP
- udp : protocole UDP
- egp : protocole EGP
- trans : informations sur les lignes de transmission
- snmp : protocole SNMP

4.2. Les variables de la MIB

Exemples de variables :

Nom	Catégorie	Sémantique
sysUpTime	system	durée depuis le démarrage
ifNumber	interfaces	nombre d'interfaces d'accès
ifMtu	interfaces	MTU(maximum transfer unit) d'une interface d'accès
ipInReceives	ip	nombre de datagrammes reçus
ipFragsOKs	ip	nombre de fragments correctement reçus
ipRouteTable	ip	table de routage
tcpRtoMin	tcp	durée minimale du temporisateur de retransmission
udpInDatagrams	udp	nombre de paquets UDP reçus

5. Représentation des données

5.1. Introduction

Les informations de la MIB peuvent être :
- simple : ipInReceives = [0 à 2^{32}-1]
- complexe : ipRouteTable !!!
- typée : ipAdresss (4 octets)

Les représentations existantes sont nombreuses. Par exemple les entiers :
- complément-à-1, complément-à-2, …
- sur un octet ou plusieurs, sur un mot, …
- longueur fixe ou variable, …

☞ ASN-1 (Abstract Syntax Notation : X409)
définit le nom, le type des variables et leur représentation (codage)

Exemple : (information sur les interfaces d'une station)

```
ipAddrTable ::= SEQUENCE OF IpAddrEntry
IpAddrEntry ::= SEQUENCE {
   ipAdEntAddr   IpAddress, -- @IP de l'interface
   ipAdEntIfIndex   INTEGER, -- numéro de l'interface correspondante
   ipAdEntNetmask   IpAddress, -- "netmask" associé
   ipAdEntBcastAddr   IpAddress, -- @IP de diffusion
   ipAdEntReasmMaxSize   INTEGER(0...65535) -- longueur max. du datag.
}
```

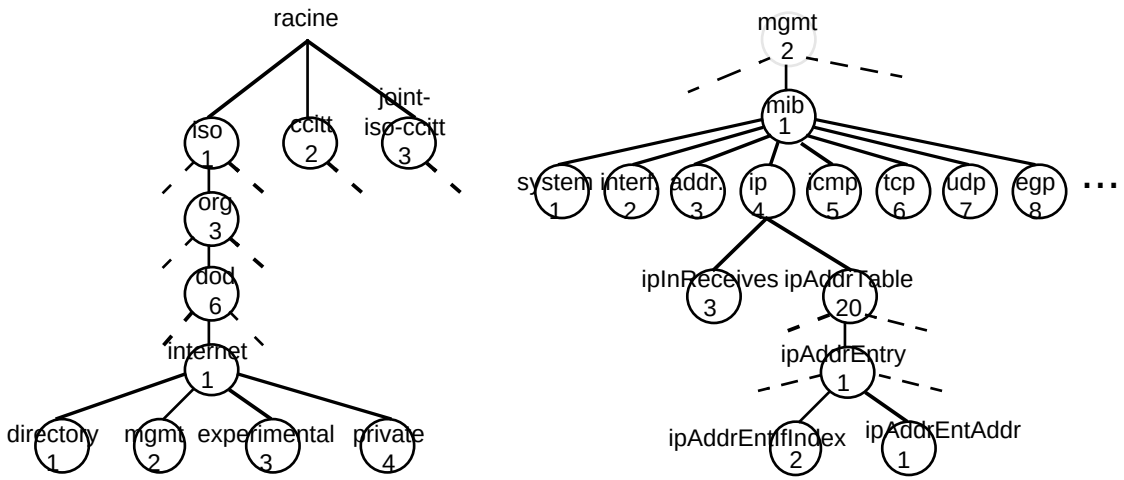
5.2. Dénomination des variables

Pour toutes les variables actuelles et futures,

Pour tous les objets (normes de protocole, compteurs d'événements, paramètres de configuration, …)

Délégation d'attribution (efficacité) et maintien de la cohérence

☞ un espace global, arborescent : MIT (“Management Information Tree”)



⇐ les différentes catégories

L’OID (“Object identifier”) :

iso.org.dod.internet.mgmt.mib.ip.ipAddrTable

☞ 1.3.6.1.2.1.4.20.0

5.3. Définition des variables

Son type :

- en syntaxe ASN-1 : soit un type universel, soit un type défini dans le rfc 1155

Le type d'accès autorisé :

- 4 types : read-only, read-write, write-only, non-accessible.

Son status d'existence :

- "mandatory, optionnal, deprecated, obsolete"

La description textuelle de la variable.

Sa dénomination.

❑ Exemple :

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 …"

::= {system 1}

6. Le protocole SNMP

Le protocole SNMP : “Simple network management protocol”

Echange d’informations sur les variables de la BdD

Opérations sans mémoire (“selfcontent message”) :

☞ - stabilité, simplicité, flexibilité

Opérations atomiques :

☞ - cohérence,

SNMP operations :
- get.Request : demande d'obtention de la valeur d'une variable
- get-next.Request : demande de la valeur de la variable suivante (non-explicitement nommée)
- get.Response : réponse à une demande
- set.Request : demande de stockage d'une valeur dans une variable
- trap : notification d'évènement

Les messages SNMP utilise le protocole UDP, numéros de port 162 (trap) et 161 (autres)

6.1. Le format général des messages SNMP

☞ en ASN-1 !

```
SNMP-message ::= SEQUENCE {
    version INTEGER {
                version-1 (0)
   },
    community OCTET STRING, -- l'ensemble des administrateurs ayant accès à l'agent
     data ANY                          -- le PDU
}
```

```
SNMP-PDUs ::= CHOICE{
      get-request GetRequest-PDU,
       get-next-request GetNextRequest-PDU,
       get-response GetResponse-PDU,
       set-request SetRequest-PDU,
       trap Trap-PDU
}
```

Le format des messages SNMP (suite)

```
GetRequest-PDU ::= [0] IMPLICIT SEQUENCE {
        request-id RequestID,
        error-status ErrorStatus,
        error-index ErrorIndex,
        variable-bindings VarBindList
}
```

avec RequestID : un type entier sur 4 octets,

☞ association entre demande et réponse

ErrorStatus et ErrorIndex : deux types d'entiers sur un seul octet,

- initialisés à zéro lors d'une requête,

☞ informe sur le déroulement de la demande

VarBindList : le type liste de noms de variable,

- couple nom et valeur de la variable (“null” pour une requête)

☞ liste des variables dont on veut obtenir la valeur

6.2. Exemple d'encodage d'une variable

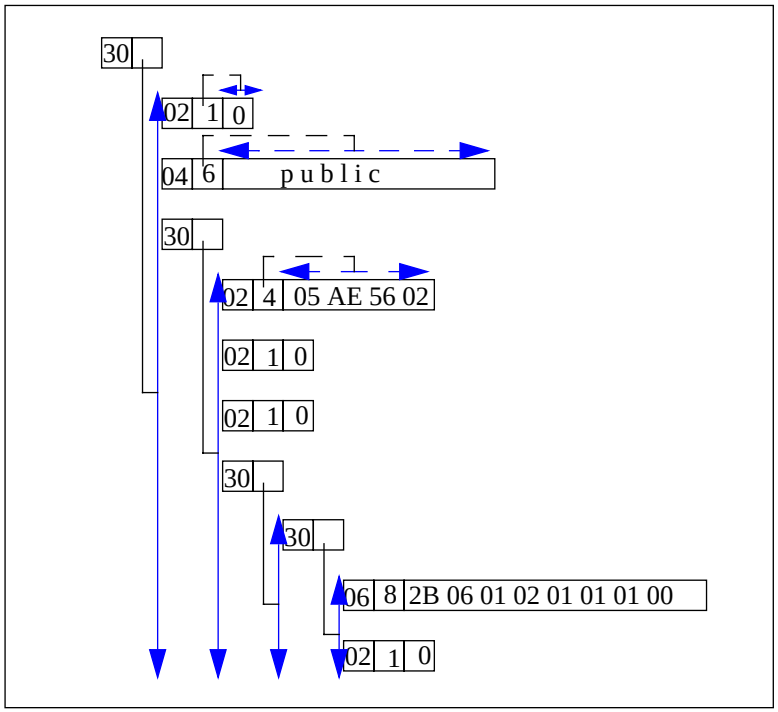
☞ encodage de GetReq(<SysDescr>) !

nota : le suffixe .0 référence l'instance de la variable dont le nom correspond au préfixe.

Règle d'encodage BER ("Basic Encoding Rules") de type TLV (type, longueur, valeur)

```
SEQUENCE  len=41
    30          29
  INTEGER    len=1      value              -- version (type prédéfini : entier)
     02          1          00
  STRING     len=6      value              -- community : "public"
     04          6         "public"
  GetReq_PDU  len=28                        -- type dépendant du contexte spécifique
     A0          1C                         -- de l'application, ici : SNMP
       INTEGER    len=4     value_request_id
            02          4         05 AE 56 02
       INTEGER    len=1     value_errror_status
            02          1         00
       INTEGER    len=1     value_error_index
            02          1         00
       SEQUENCE   len=14                    -- une liste …
            30          0E
            SEQUENCE   len=12               -- …. de couples
                 30          0C
                 Object_id      len=8   value_object=1.3.6.1.2.1.1.1.0   -- Les deux 1er labels
                      06          8       2B 06 01 02 01 01 01 00        -- sont encodés ensemble
                 NULL           len=0
                      05          0
```

La structure de données correspondante :



7. Conclusion

7.1. Généralités

SNMP

Protocole simple, minimal et sans mémoire

3 opérations :
- obtenir une valeur
- modifier la valeur d'une variable
- notifier l'apparition d'un évènement

MIB

Base de données générale identifiant les objets à l'aide d'une arborescence

Les objets sont représentés à l'aide d'ASN-1 et encodés par BER (cf. XDR)

7.2. Améliorations

☞SNMP v2

Amélioration de la sécurité :

. authentification

. protection

Communication entre administrateurs SNMP

. opération : InformRequest

Accès simultanée à plusieurs variables de la MIB

. opération : GetBulkRequest

☞ optimisation, structure de taille inconnue