

# TD N° 02 : SNMP

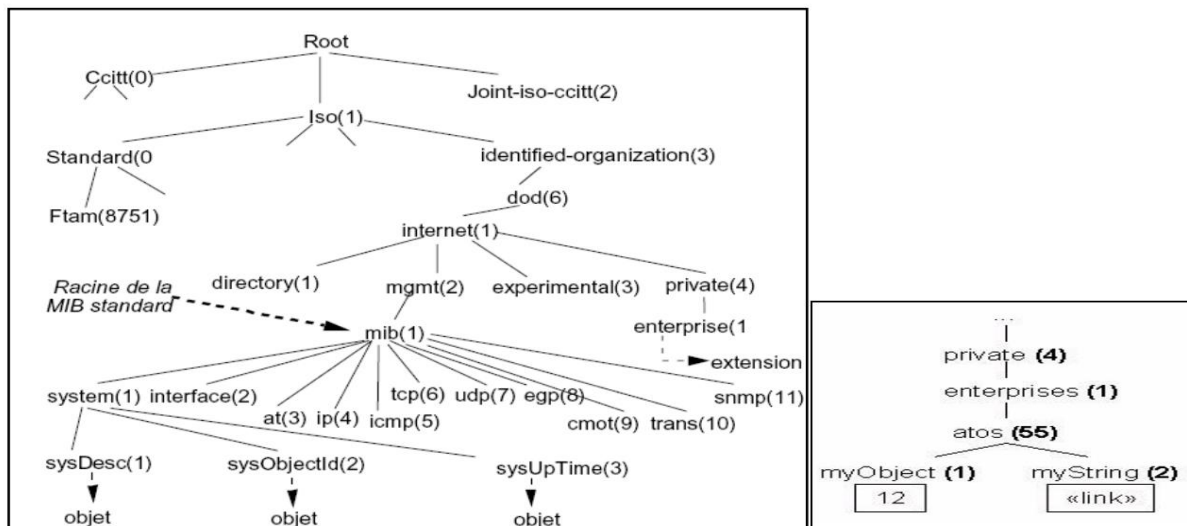
## Exercice n 01 :

1. Pourquoi est-il nécessaire de définir un protocole d'administration de réseau ?
2. Qu'est-ce qu'un agent SNMP ?
3. Quel est le protocole de transport utilisé ? quelle est la différence entre ce protocole et le protocole TCP/IP ?
4. Comment un objet de la base est-il accessible ?
5. Répondre par vrai ou faux aux assertions suivantes en justifiant votre réponse.
  - 1- Un manager qui a les droits de lecture et d'écriture ne peut pas modifier la valeur de n'importe quel objet de la MIB
  - 2- Tout objet de la MIB est lui associé une valeur.
  - 3- SNMP peut utiliser le même message pour lire la valeur d'un objet et modifier la valeur d'un autre objet puisqu'il permet un accès multiple.
  - 4- La taille d'un message SNMP est fixe
  - 5- Dans SNMPv2, les messages GET, GETNEXT et SET ne sont envoyé que par le manager.
  - 6- La station d'administration communique directement avec les objets administrés.
  - 7- SNMP peut utiliser le même message pour lire plusieurs objets de la MIB.

## Exercice n 02 :

On considère une partie de la MIB d'un agent SNMP représentée ci-dessous.

- 1- Donner les OID des objets « myString » et « myObject »
- 2- Le manager désire récupérer la valeur de l'objet « myString » puis modifier la valeur de l'objet « myObject » à 20. Donner, l'échange des messages ( entre le manager et l'agent) nécessaires pour réaliser les deux opérations en indiquant les attributs pertinents de chaque message.
- 3- Peut-on modifier la variable Root par SNMP? justifier.



## Exercice 3 :

Soient les deux messages SNMP suivants représentant une requête et sa réponse :

- 1- Déterminer la version de SNMP utilisé.
- 2- Préciser le type de la requête (simple ou multiple). Expliquer.
- 3- Préciser le « PDU type » du premier message (GET ou GET-NEXT ou SET). Expliquer.
- 4- Quelle est la signification de la valeur 11915034 dans le deuxième message? a-t-elle une relation avec le texte qui le suit ? Expliquer.

```

Simple Network Management Protocol
  Version: 1
  Community: public
  PDU type: G .....
  Request Id: 0x25
  Error Status: NO ERROR
  Error Index: 0
  Object identifier 1: 1.3.6.1.4.1.9.2.1.58.0 (SNMPv2-SMI::enterprises.9.2.1.58.0)
  Value: NULL
  Object identifier 2: 1.3.6.1.4.1.9.2.1.57.0 (SNMPv2-SMI::enterprises.9.2.1.57.0)
  Value: NULL
  Object identifier 3: 1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
  Value: NULL
Simple Network Management Protocol
  Version: 1
  Community: public
  PDU type: RESPONSE
  Request Id: 0x25
  Error Status: NO ERROR
  Error Index: 0
  Object identifier 1: 1.3.6.1.4.1.9.2.1.58.0 (SNMPv2-SMI::enterprises.9.2.1.58.0)
  Value: INTEGER: 16
  Object identifier 2: 1.3.6.1.4.1.9.2.1.57.0 (SNMPv2-SMI::enterprises.9.2.1.57.0)
  Value: INTEGER: 16
  Object identifier 3: 1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
  Value: Timeticks: (11915034) 1 day, 9:05:50.34

```

Soient la requête SNMP suivante:

```

Simple Network Management Protocol
  Version: 1
  Community: public
  PDU type:
  Request Id: 0x25
  Error Status: NO ERROR
  Error Index: 0
  Object identifier 3: 1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
  value: 15246874

```

- 5- Déterminer la version de SNMP utilisé.
- 6- Préciser le type de la requête (simple ou multiple). Expliquer?
- 7- Préciser le « PDU type » (GET ou GET-NEXT ou SET). Expliquer ?
- 8- A votre avis, que sera la réponse de l'agent (en vos propres mots) ? Expliquer?

#### **Exercice 4 :**

En vous aidant des formats des messages SNMP et des arbres et MIB, analyser les messages SNMP ci-dessous pour répondre aux questions suivants :

- 1- Préciser la version de SNMP utilisé et le nom de communauté.
- 2- Quels sont les messages envoyés par le manager. Identifier, dans ces messages, l'information demandée par le manager.
- 3- Quel sont les messages envoyés par l'agent. Identifier, dans ces messages, l'information envoyée par l'agent.
- 4- Si applicable, identifier les messages réponses aux messages envoyés par le manager. Expliquer ?

<p><b>Message1:</b>  SNMP: len: 38 version: int(1) 0x00 comm: string(6) «public» type: GET-NEXT  req-id: int(2) 0x5e31 error: int(1) 0x00 error-index: int(1) 0x00  var: obj(8) 1 3 6 1 2 1 2 1 val: empty(0)</p>	<p><b>Message2:</b>  SNMP: len: 40 version: int(1) 0x00 comm: string(6) «public» type: RESPONSE  req-id: int(2) 0x5e31 error: int(1) 0x00 error-index: int(1) 0x00  var: obj(7) 1 3 6 1 2 1 2 1 0 val: 0x06</p>
<p><b>Message3:</b>  SNMP: len: 178 version: int(1) 0x00 comm: string(6) «public» type: GET-NEXT  req-id: int(2) 0x00a2a2 error: int(1) 0x00 error-index: int(1) 0x00  var: obj(9) 1 3 6 1 2 1 2 2 1 1 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 2 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 3 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 4 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 5 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 6 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 7 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 8 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 9 val: empty(0)  var: obj(9) 1 3 6 1 2 1 2 2 1 10 val: empty(0)</p>	<p><b>Message4:</b>  SNMP: len: 219 version: int(1) 0x00 comm: string(6) «public» type: RESPONSE  req-id: int(2) 0x00a2a2 error: int(1) 0x00 error-index: int(1) 0x00  var: obj(10) 1 3 6 1 2 1 2 2 1 1 1 val: int(1) 0x01  var: obj(10) 1 3 6 1 2 1 2 2 1 2 1 val: string(9) «Ethernet0»  var: obj(10) 1 3 6 1 2 1 2 2 1 3 1 val: int(1) 0x06  var: obj(10) 1 3 6 1 2 1 2 2 1 4 1 val: int(2) 0x05dc  var: obj(10) 1 3 6 1 2 1 2 2 1 5 1 val: gauge(4) 0x00989680  var: obj(10) 1 3 6 1 2 1 2 2 1 6 1 val: string(6) «*****»  var: obj(10) 1 3 6 1 2 1 2 2 1 7 1 val: int(1) 0x01  var: obj(10) 1 3 6 1 2 1 2 2 1 8 1 val: int(1) 0x01  var: obj(10) 1 3 6 1 2 1 2 2 1 9 1 val: time(2) 0x0420  var: obj(10) 1 3 6 1 2 1 2 2 1 10 1 val: counter(4) 0x6b055aa0</p>

**Exercice 5 :**

La taille moyenne d'un message SNMP étant supposée de 100 octets, quelle est l'influence de la consultation de 100 objets SNMP à travers un lien WAN si la période de polling est fixée à 10 s et que le débit du lien est de 64 kbit/s.

**Exercice 6 :**

Un analyseur de protocole a relevé la trace suivante, il s'agit d'une trame Ethernet, contenant une requête SNMP.

```

Captured at: +00:03.004
Length: 100      Status: Ok
OFFST DATA                                           ASCII
0000: 00 A0 24 BD 75 DB 08 00 02 05 2D FE 08 00 45 00  ..$.u.....-...E.
0010: 00 52 3C EF 00 00 1C 06 A4 FF 80 00 64 00 D0 80  .`<.....d...
0020: 08 29 0A CF 00 A1 47 A8 BA 20 01 A3 96 14 50 18  .)....+G.. ....P.
0030: 20 00 72 D4 00 00 30 28 02 01 00 04 06 70 75 62  .r...0c.....pub
0040: 6C 69 63 A0 1B 02 04 03 05 52 AE 02 01 00 02 01  lic.....
0050: 00 30 0D 30 0B 06 07 2B 06 01 02 01 01 03 05 00  .0.0...+.....
0060: 9F 59 6E FC                                           .Yn.

```

Les règles de codage des SNMP\_PDU sont données ci-dessous.

Message SNMP DEFINITION ::=

BEGIN

MESSAGE\_SEQUENCE ::= SEQUENCE

{ version Integer -- Version = 0 pour MIB 1

community Octet String -- "Mot de passe" de longueur variable

data Any -- Données du message

}

PDU ::= CHOISE

{ gest\_request GetRequest\_PDU,

gest\_next\_request GetNextRequest\_PDU,

get\_response GetResponse\_PDU,

set\_request SetResponse\_PDU,

trap Trap\_PDU

}

END

GetRequest\_PDU ::= [0] Implicit Sequence

{ request\_ID Request\_ID, -- Identification du couple question/réponse

error\_status ErrorStatus, -- Toujours à 0

error\_index Error\_Index, -- Toujours à 0

variable\_bindlist VarBindList -- Liste OID et valeurs dans PDUs réponse

}

- Décodez cette trame, et analysez les champs de la SNMP\_PDU transportée.