Administration Réseau-système
SNMPv1, SNMPv2, SNMPv3 et HTTP

Yves.Bertsch@lapp.in2p3.fr
Frederic.Stmarcel@lapp.in2p3.fr

Résumé : *Cet article développe les solutions actuelles de l'administration réseau, en insistant particulièrement sur SNMP. Après un rappel des défauts de jeunesse de SNMP, on exposera SNMPv3, protocole mature, muni de tous les outils de sécurité requis en milieu ouvert. SNMPv3 sera sans doute incontournable dans la gestion des grands réseaux, exigeant efficacité et sécurité. Au dessous d'une certaine taille de réseaux, des solutions très diversifiées, souvent orientées objet, plus ou moins intégrées dans le WEB seront offertes à l'utilisateur. Ces approches, qui privilégient l'intelligence et l'initiative dans l'appareillage se différencient de l'approche scalaire classique de SNMP. L'article cite quelques unes de ces techniques de gestion alternatives et parfois complémentaires à SNMP.*

Introduction

SNMP s'inscrit dans la marche vers les standards, approche nécessaire dans un monde de constructeurs diversifiés. La démarche pragmatique de TCP/IP a longtemps été opposée à la réputation de complexité et de lourdeur de mise en œuvre des produits de l'OSI.

L'IETF et OSI ont mené à terme une administration réseaux dont les cahiers des charges étaient proches, mais avec des techniques différentes. La place prépondérante de SNMP est elle définitive ? La complexité de SNMPv3 ne rappelle t elle pas les défauts des produits OSI ?

Deux ans bientôt après le bouclage des définitions de l'IETF sur SNMPv3, qu'en est il des produits SNMPv3 ? Ce protocole répond-il aujourd'hui réellement à une demande ? L'émergence des techniques orientées WEB ne menace-t-elle pas l'approche classique de l'administration réseau ? Que faut-il attendre de la puissance logicielle implantée dans les équipements ? Autant de questions qui nous interpellent, la nécessité de prendre en compte l'administration réseaux (et système) n'étant plus à démontrer.

Nous essayerons de mettre en évidence les points importants de l'évolution des techniques d'administration SNMP en illustrant avec un aperçu des résultats obtenus sur des agents cisco3640 SNMPv1/v2c et v3 (rares en équipements). Ces évaluations ont été menées avec des gestionnaires SNMPv3 sur LINUX et NT . Des agents SNMPv3 tournant sur LINUX et NT ont été également testés. Les réflexions en cours actuellement sur les évolutions de SNMP seront abordées. Les techniques alternatives ou complémentaires à SNMP, le plus souvent à base de gestion par le WEB, seront succinctement présentées ; le sujet justifierait à lui seul une présentation complète.

Les bases de SNMP sont supposées connues et on insistera sur l'apport de SNMPv3. Toutefois des compléments sur SNMP ainsi que des résultats plus détaillés de développements et tests sont rappelés en ANNEXE et sont disponibles sur les sites WEB de JRES et du LAPP. L'emploi de sigles ou d'expressions anglaises (plus compactes) pouvaient difficilement faire l'objet d'une traduction en pied de page. On trouvera un glossaire à la fin de l'article.

1. Historique, buts et bilan de SNMPv1

L'origine de SNMP remonte à la fin des années 80 et coïncide avec l'accélération de la croissance mondiale de TCP/IP. SNMP, sous l'appellation SGMP (G pour Gateway), est d'abord une réponse à la gestion des routeurs IP qui doivent écouler un trafic en forte expansion. A cette époque, la concurrence entre OSI et TCP/IP est encore vive et cette situation explique en partie les bases sur lesquelles les concepteurs de SNMP vont s'appuyer. Il s'agit de laisser ouverte une convergence possible avec CMIP/CMIS, protocole d'administration de OSI qui est alors en développement. Ainsi les choix de ASN-1 pour la syntaxe abstraite (écriture des MIBs) et BER pour l'encodage sur le réseau, l'intégration dans l'espace de nommage de l'ISO pour la classification des variables qui allaient constituer la MIB SNMP illustrent cette démarche.

L'approche pragmatique de l'IETF a conduit rapidement à un protocole volontairement simpliste dont le domaine d'application a très vite débordé du cadre de départ (les routeurs) pour s'intéresser à l'ensemble des constituants du réseau, puis du système par la suite.

L'affirmation de l'Internet comme standard de fait a consacré le divorce avec OSI (échec de CMOT : CMIP over TCP) et permet à l'IETF de mettre en chantier la seconde mouture de SNMP, appelée à améliorer les insuffisances du cadre d'administration et à introduire les outils indispensables de sécurisation (authentification

et cryptage).

Si la nécessité de se doter d'un outil de contrôle de la croissance de l'Internet a été un élément déterminant dans le démarrage de SNMP, la motivation collective à s'engager dans la voie des standards a également beaucoup compté.

Une décision importante fut d'allouer aux constructeurs une branche privée dans l'espace de nommage, leur permettant de traduire dans le cadre du standard toute leur spécificité. La MIB généraliste de départ (MIB-I, puis MIB-II) était en effet très pauvre et les particularités des équipements réseaux n'étaient pas encore développées dans la MIB standard.

Avant de développer SNMPv3, sujet principal ici, il est utile de faire un bilan de SNMP à ce jour, bilan en fait de SNMPv1 et SNMPv2c (qui est en gros SNMPv1 avec les améliorations du cadre d'administration SMIv2, qui sera évoqué plus loin).

1.1 Rappel SNMPv1

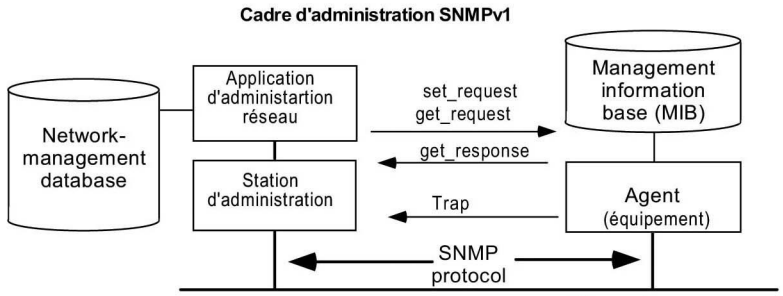


Figure 1

SMIv1 : RFC1155-1212
Définition du cadre du langage d'écriture des MIBs en ASN.1

Protocole SNMP : RFC1157 :
Les échanges entre entités SNMP
Protocole Data Units (PDU)
Get_request (lecture), Trap (alerte), Set_request (écriture), …

Figure 1

Contrairement à CMIP (peu diffusé) qui a adopté d'entrée une approche orientée objet, SNMP a choisi une approche scalaire. La complexité est concentrée dans le gestionnaire, l'agent se contentant principalement de répondre aux requêtes du gestionnaire . L'agent peut cependant aussi envoyer spontanément des alarmes ou alertes (TRAP). Ce schéma répond aux possibilités de l'époque , la puissance CPU de l'agent étant à l'origine limitée.

1.2 L'espace de nommage

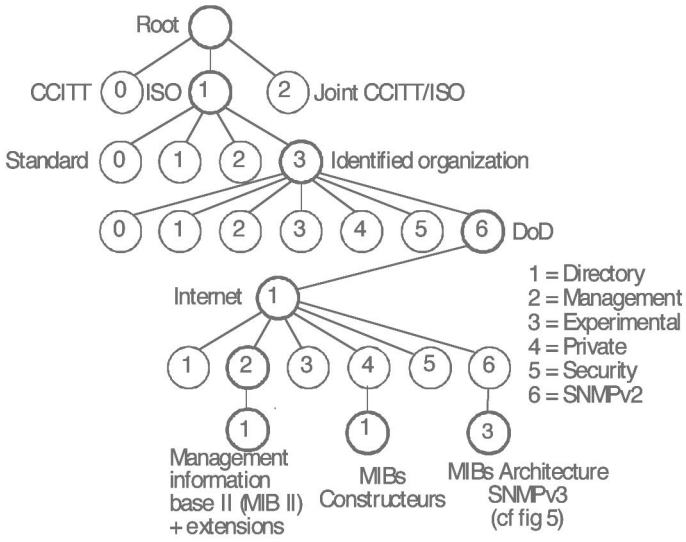
Avant SNMP, les équipements réseau (hubs, ponts) possédaient déjà un mécanisme de dialogue (rs232, telnet) à des fins d'administration. La nature des données explorées, bien que similaires pour les paramètres courants, dépendaient de l'implémentation des constructeurs. En particulier, chaque constructeur gérait sa propre classification des variables jugées utiles, selon sa perception.

Le corollaire de cette situation était évidemment que la fonction administration des équipements émanait du constructeur avec tous les inconvénients que cela implique.

Avec SNMP, toute donnée accédée dans un équipement (agent) correspond à une variable définie et répertoriée par les organismes de standardisation (IETF, IANA …). Pour SNMP, le choix stratégique de l'espace de nommage de l'ISO commandait d'adopter le langage de définition des variables (ASN.1), l'encodage sur le réseau (BER). Par souci de simplification, SNMP s'est limité à un sous ensemble de ces mécanismes.

La structuration arborescente de l'espace de nommage est classique. On parle de « base de données virtuelle ». Un équipement est accessible en tant que nœud IP (processus agent), lequel autorise (sous contrôle) l'accès aux paramètres le décrivant par une MIB. Ces objets, **administrativement** gérés, sont **nommés** en donnant le parcours complet depuis la racine :

1.3.6.1.2.1.x.y….pour les variables de la MIB standard,
1.3.6.1.4.1.x.y… pour les variables MIB constructeurs, x étant No attribué au constructeur
1.3.6.1.6.3.x.y… pour les variables traduisant l'architecture SNMPv3



Espace de nommage de l'ISO

Branchement réservé à SNMP : 1.3.6.1

Les chemins les plus utilisés sont :

1.3.6.1.2.1….Ensemble des MIB du standard (MIB-2, extensions…) développées par l'IETF

1.3.6.1.4.1…. espace des MIBs constructeurs

1.3.6.1.6.3…. snmpv2, en fait contient les MIBs associées aux sous systèmes de l'architecture SNMPv3 (framework, USM, Coex, VACM…)

Figure 2

1.3 Aspects du langage ASN.1 (Abstract Syntax Notation Number 1) et BER (Basic Encoding Rules)

La compréhension de ASN.1 est incontournable pour quiconque doit travailler dans l'environnement SNMP. Qu'il s'agisse d'approfondir la fonction de telle variable d'une MIB standard ou privée, de modifier une partie de MIB qui ne passe pas la compilation, de consulter une RFC sur le protocole ou la cadre d'administration, la connaissance des mécanismes de ce langage est indispensable.

Il s'agit d'un langage, à usage de lecture humaine. Pas d'initialisation de variables, pas de dimensionnement de tableaux, tout au plus des restrictions du champ de valeurs possibles. ASN.1 doit être transposé en langage exécutable (C ou assembleur) pour intégrer le binaire que constitue l'agent (transparent à l'utilisateur). On retrouve partiellement les caractéristiques de l'approche objet : une variable (de la MIB) est créée par invocation d'une « macro » (terme impropre parfois appelée « construct », Perkins [12]. La puissance mais aussi le danger de ASN.1 (on peut indéfiniment enrichir la grammaire) est bien soulignée par T.Marshall ROSE [9].

SMIv1 et SMIv2 (Structure of Management Information : Cadre d'administration)
Ce qu'on appelle SMIv1, c'est les définitions du langage ASN.1 adapté à SNMP, c'est à dire les types primitifs universels ISO simples (INTEGER, OCTET STRING…) et construits (SEQUENCE : concaténation de types simples), ainsi que les types spécialement créés pour SNMP (ipAddress, Counter…). C'est aussi les règles de grammaire (MACRO) pour créer les objets qui vont peupler les MIBs. La simplification un peu excessive et le manque de prévision (compteurs limités à 32 bit) a demandé une nouvelle version appelée SMIv2.

La manipulation des tables est un peu déroutante, assez lourde et inélégante dans SNMPv1. Des améliorations sensibles ont été apportées dans SNMPv2, mais les tables restent un point controversé dans SNMP (pas de commande directe pour obtenir le contenu d'une table). Quelques précisions sont apportées en ANNEXE.

1.4 Bilan de SNMPv1/v2C :

SNMPv2 en version initiale (1993) devait corriger d'une part les imperfections de SNMPv1 (SMIv1 trop étriqué) et surtout apporter le volet sécurité. En effet dans SNMPv1, la sécurité repose sur la connaissance mutuelle (entre agent et gestionnaire) d'une chaîne de caractère (la communauté) insérée dans le message. Cette communauté n'étant pas cryptée, elle est interceptable sur le réseau.

Le premier point a été atteint (SMIv2), mais le volet sécurité n'a pas rencontré le consensus. Ce standard, SNMPv2p (p pour party based) reposait sur un dialogue entre entités appelées « party », et devait déboucher sur un nouveau format de message (qui n'a pas abouti). Pour ne pas perdre le bénéfice des avancées du SMIv2, on a gardé l'enveloppe du message SNMPv1 (communauté) et utilisé le SMIv2 dans les MIB; l'enrichissement du protocole (nouvelles Unités de protocole : PDU) a été également retenu. C'est ce qu'on appelle SNMPv2c (C pour community), classé « expérimental ». On fera donc un bilan global SNMPv1/v2C.

1.4.1 Points négatifs

- Reproche principal : manque de sécurité (communauté circule en clair sur le réseau)
- Manipulation des tableaux lourde et inefficace (sensiblement améliorée par le SMIv2).
- Protocole inefficace (pas de transfert de masse dans V1), corrigé dans V2c (PDU getbulk)
- Alertes (TRAP) non acquittées dans SNMPv1, corrigé dans v2C avec la PDU de notification INFORM (acquittée) en plus de TRAP(non acquitté)
- Entiers limités à 32bit (V1), insuffisant pour les compteurs (corrigé dans SMIv2 avec compteur 64 bit)
- Pas de réelle possibilité de limitations sélectives d'accès à la MIB de l'agent
- Nommage des variables trop long (tout le parcours 1.3.6.1.2.1.., surtout pour les éléments de tableaux), aggravé par l'encodage BER. Impact perceptible sur la bande passante du réseau, en particulier en liaison géographique de faible débit (64 Kbit).
- Mécanisme atomique de V1 (une erreur dans la requête annule toute la requête), corrigé par V2 par le mécanisme des « exceptions » (donne le maximum de réponses en cas d'erreur sur une variable)
- Manque de coordination entre gestionnaires , la MIB M2M (manager to manager) réglementait ce type d'échange dans snmpv2 sécurisé (1993) qui n'a pas été retenu.

1.4.2 Points positifs :

- Standard bien accepté dans l'ensemble, tous les constructeurs l'implémentent dans leur équipement, les MIB sont écrites systématiquement en SMIv2)
- Couverture vaste des MIB décrivant les équipements les plus divers du réseau
- Une certaine compatibilité d'approche avec le monde OSI (ASN.1, espace de nommage)
- Grand choix de gestionnaires (du petit browser à la puissante plate-forme HPOV)
- Le SMIv2 apporte des fonctionnalités nouvelles dans la manipulation des tables
  AUGMENT : permet de prolonger (ajout de colonnes) une table existante
  Type ROWSTATUS, addition/suppression dynamique de rangées dans les tables.

Cette dernière fonctionnalité est utilisée de façon systématique dans les tables décrivant les sous systèmes du nouveau cadre d'administration de SNMPv3.

On constate qu'un certain nombre d'imperfections ont pu être corrigées par l'adoption du SMIv2 avec SNMPv2C, version d'attente faute de mieux, qui laissait de coté le volet sécurité.

1.5 Volet sécurité reporté

Le rejet du volet sécurité (1993) SNMPv2p (P pour party based, l'entité communicante SNMP s'appelant « party »), a tenu essentiellement à un mécanisme de manipulation de clés (authentification et cryptage) impraticable.

Deux standards concurrents allaient alors s'affronter pendant deux années jusqu'à début 97, SNMPv2U et SNMPv2*. Ces deux standards avaient en fait déjà adopté le mécanisme de clés « localisées » qui sera retenu dans SNMPv3. Il était donc envisageable de faire une synthèse de ces deux standards, une fois les esprits calmés, condition impérative pour sauver le protocole SNMP.

2. Version SNMPv3 Aboutissement actuel – (achevé fin 1999, déploiement en cours)

Le groupe de travail (1997) chargé de concilier les approches concurrentes, a redéfini complètement la structure du cadre d'administration tout en posant en principe l'acquis de l'essentiel du SMIv2 et l'enrichissement du dialogue entre entités SNMP (nouvelles unités de données de protocoles, PDU Getbulk, Inform, Report).

2.1 Cahier des charges de SNMPv3

1) Introduire les outils de sécurisation (authentification, cryptage, contrôle du timing).

Défi : satisfaire à la fois les besoins en administration lourde (grands réseaux, sécurité, gestion à distance sécurisée, y compris des clés …) et les besoins simples d'administration d'un petit réseau local sans subir une lourdeur inutile.

2) Rendre possible une évolution séparée des différentes composantes de la nouvelle architecture. La nouvelle approche est nécessairement modulaire, afin de satisfaire cet objectif. Elle a aussi pour avantage de ne pas figer la structure du message, sujet sensible.

L'application SNMP, globale auparavant, se divise à présent en deux sous ensembles (fig 4, pour un agent) :

• snmpEngine : machine (ou moteur) SNMP, chargé de l'instrumentation du protocole ;
• Applications : *Command Responder* désigne l'ancienne fonction agent, Notification regroupant TRAP et INFORM *(acquittée)*. Proxy-forwarder est l'application qui permet de traduire un message d'une version SNMP dans une autre version.

Les applications s'appuient sur ce *moteur* SNMP, pour échanger les unités de données de protocole (PDU). Ci dessous la représentation d'une entité agent. L'entité gestionnaire se déduirait aisément (comme indiqué sur la figure 4). On notera que la structure modulaire permet éventuellement de faire coexister applications agent (Command Responder) et gestionnaire (Command Initiator), un gestionnaire pouvant se comporter en agent vis à vis d'un autre gestionnaire.

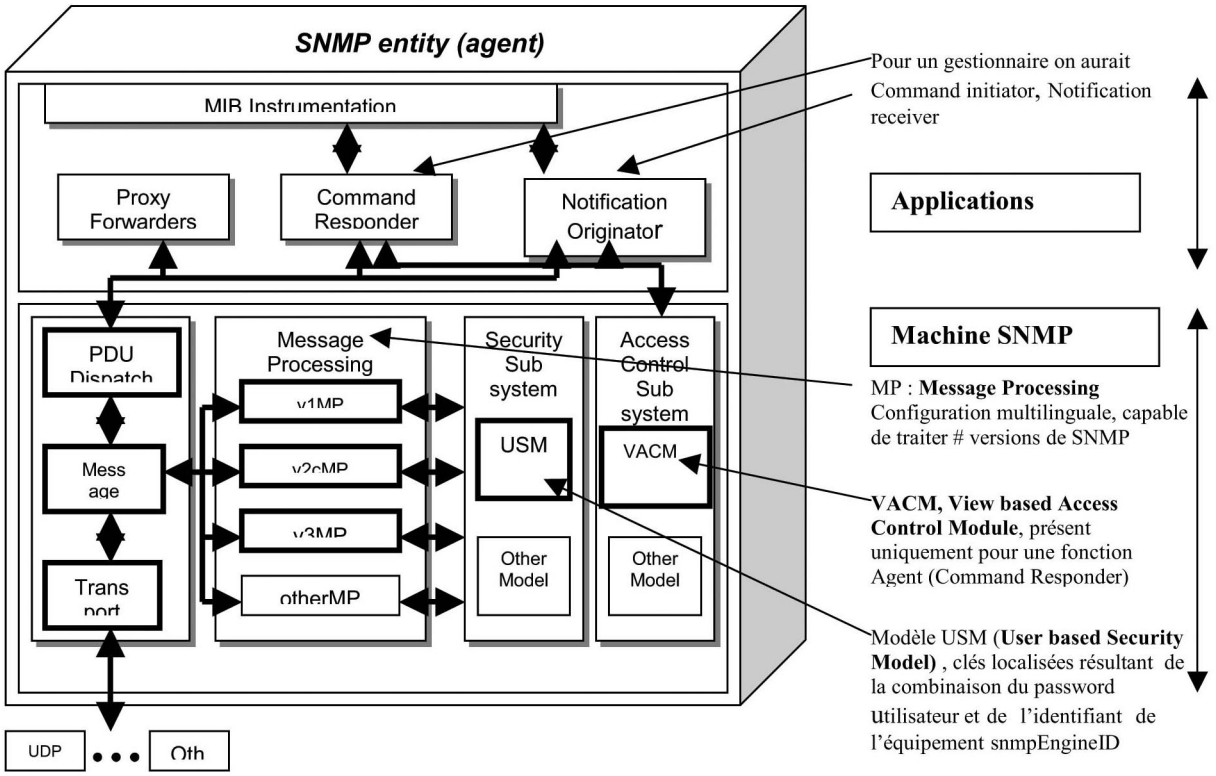
La démarche du groupe de travail, faire la synthèse entre 2 versions concurrentes, n'était pas simple vu la tension qui régnait dans la communauté SNMP. Il y a eu beaucoup de polémiques avant que les esprits ne se calment.

Toutes ces discussions ont retardé l'aboutissement du projet, mais ont obligé les auteurs à une rigueur dans la formulation des concepts et dans la rédaction des documents. Cela limitera sans doute les interprétations différentes des documents. Mais le retard accumulé au fil des années a lassé utilisateurs et constructeurs qui ont entrepris de se tourner vers d'autres techniques, principalement axées sur le WEB. On verra plus loin succinctement les solutions offertes sur le marché aujourd'hui.

SNMPv3 est toutefois aujourd'hui une réalité. Les concepts développés ont enrichi SNMP et doivent permettre un déploiement sur mesure, avec des outils masquant la complexité de la nouvelle architecture.

2.2 Description de l'architecture de SNMPv3

Figure 4 (Entité SNMP : Machine SNMP + Applications)



Les services entre modules dans une entité SNMP sont définis en terme de primitives avec paramètres. Ces mécanismes sont dépendants de l'implémentation. On utilise le terme de **ASI** , Abstract Service Interface pour qualifier le mécanisme de services entre modules.

Dans la nouvelle architecture SNMP le sous système *Dispacher* est la plaque tournante de l'entité SNMP, il échange (émet ou reçoit des primitives) avec :

• Le réseau pour les messages
• Les applications transmettant/réceptionnant les unités données de protocoles (PDU), c'est à dire les données.
• Le module (MP : Message processing)) de traitement du message (V1, V2c ou V3).

Le module de traitement du message (MP) émet une primitive à destination du module de sécurité (USM). Si

l'entité SNMP émet un message, USM sécurise le message à transmettre selon les directives que MP a reçu de l'application (paramètres de temps, remplissage éventuel des champs d'authentification, de cryptage). Dans le cas d'un message reçu, les contrôles de sécurité sont effectués selon les directives du traitement message (MP).

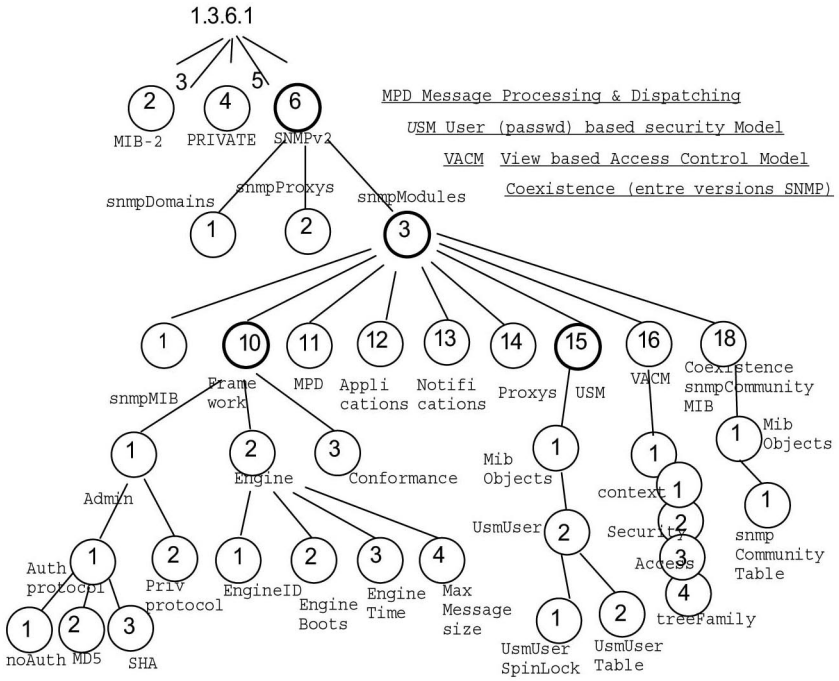
Dans ce cas, si les paramètres de sécurité satisfont aux critères attendus, MP repasse la main à l'application pour traitement des données (PDU). Dans le cas d'un agent (*Command Responder*), la primitive d'appel au contrôle d'accès par VACM est initiée par l'application (ASI : isAccessAllowed, paramètres). Il s'agit de vérifier que l'accès à la MIB de l'agent est conforme à ce qui est permis pour le demandeur (« Principal ») en question. Le mécanisme de contrôle d'accès est décrit en détail plus loin .

Il est à noter que le sous système de contrôle d'accès (VACM), n'est présent que sur un *Command Responder. Important* : sur un agent *(Command Responder*) supportant SNMPv3, VACM peut imposer des restrictions d'accès à des commandes SNMPv1,v2C. En effet une translation directe *(mapping)* de la communauté en paramètres interprétables par VACM permet cette opération (community-security-Model de SNMP-COMMUNITY-MIB, RFC 2576) qui se rattache aux problèmes généraux de la coexistence des versions SNMP.

2.3 Espace de nommage pour les objets (les plus courants) de l'architecture SNMPv3

Cet espace de nommage rassemble l'ensemble des objets décrits dans les RFC 2570 à 2576. qui composent l'entité globale SNMP, et qui se trouvent sous l'embranchement 1.3.6.1.6.3, sous snmpModules. On trouve donc les objets manipulés dans les sous systèmes MPD (traitement du message), USM (sécurité), VACM (contrôle d'accès pour un agent), les objets de certaines applications.

Figure 5



10 Framework
RFC 2571

11 MP Message Processing
RFC 2572

12, 13, 14 Applications
RFC 2573

15 User based securitModel
RFC 2574 (USM)

16 View based Access Control
RFC 2575 (VACM)

18 Community based Security Model (coexistence)
RFC 2576

Contenu des RFC décrivant l'architecture SNMPv3

Que faut il retenir du contenu de ces nombreuses RFC …. Il serait illusoire de penser que les RFC n'intéressent que les développeurs. Comme dans tous les sujets touchant TCP/IP, il est fréquent d'avoir à consulter les RFC pour approfondir un point précis. De plus la consultation des différentes tables aident beaucoup à la compréhension des mécanismes. Les RFC qui accompagnent le schéma des sous systèmes de L'architecture SNMPv3 (fig 5) éclairent principalement 3 points :

- Les concepts qui sous tendent les fonctionnalités des sous systèmes
- Les primitives (ASI) dans lesquelles le sous système particulier est impliqué, avec leurs paramètres IN/OUT
- Les MIB associées qui précisent le rôle des variables utilisées par le sous système.

Framework : RFC 2571 (branchement 1.3.6.1.6.3.10)

RFC généraliste qui décrit l'architecture générale modulaire de SNMPv3 , avec la nouvelle approche de l'entité SNMP, le mécanisme de sécurité à deux niveau (message avec USM, accès MIB avec VACM). Toutes les

primitives entre les sous systèmes sont introduites (détaillées dans les MIB des sous systèmes).

MPD : RFC 2572 (branchement 1.3.6.1.6.3.11) MP ou MPD pour dispaching
Cette RFC traite de l'ensemble traitement du message et aiguillage (*Dispacher*) vers réseau, applications, et module de sécurité. Ce Module traite également le format du message SNMPv3(description ASN.1) en détaillant les variables du champ entête du message (voir fig 6).

Applications : RFC 2573 (branchements 1.3.6.1.6.3.12,13,14)
Ce module décrit le mécanisme d'interaction des différentes applications avec l'aiguilleur (Dispacher) et pour les applications impliquant un accès à la MIB de l'agent, la primitive isAccessAllowed (params) à destination de VACM. Les variables associées aux applications sont décrites dans les MIB spécialisées.

USM : RFC 2574 (branchement 1.3.6.1.6.3.15)

Cette RFC rappelle l'approche sécurité et protection dans SNMPv3 (reprises de SNMPv2). Le module de sécurité doit être en mesure de fournir les outils suivants :

- Authentification et intégrité du message
- Cryptage d'une partie des informations
- Protection contre la réémission de messages retardés ou hors séquence.

Cette RFC décrit les mécanismes de sécurité énoncés ci dessus, en énonçant les outils actuels disponibles. On insiste particulièrement sur la génération de clés « localisées », qui n'imposent à l'administrateur que la gestion d'un password habituel. Les mécanismes de création/suppression d'utilisateurs, de changement de clé (à distance) sont traités dans le détail. Une table importante (usmUserTable) gère les variables associées à ces mécanismes.

VACM : RFC 2575 (branchement 1.3.6.1.6.3.16)
Pour une entité SNMP agent (Command Intiator) , cette fonctionnalité constitue un apport essentiel de la nouvelle architecture. Le mécanisme d'analyse des permissions affectées à chaque administrateur permet de contrôler finement l'accès à la MIB de l'agent. 4 tables réglementent cet accès, les MIBs associées explicitant la fonction des variables.

Coexistence and Community based Security Model : RFC 2576 (branchement 1.3.6.1.6.3.18)
Pour bénéficier des mécanismes de VACM quelle que soit la version de SNMP, il était important de prévoir une adaptation de SNMPv1 et v2C à l'environnement de SNMPv3. Si l'entité agent traite les 3 types de messages SNMP, Cette adaptation est réalisée par configuration de l'agent qui transforme une communauté en un couple de variables (securityName, securityModel), compréhensible par VACM.

2.4 Points forts du modèle :

- Incontestablement l'architecture modulaire constitue l'apport majeur de SNMPv3 et offre d'entrée une grande souplesse pour intégrer différentes options possibles de traitement. Les retombées sont multiples et permettent à la fois de prendre en compte l'existant ainsi que de promouvoir une évolution indépendante pour les différentes composantes de l'architecture. cette approche, absente de SNMPv2 avait condamné l'ensemble de l'architecture du fait de l'échec du volet sécurité. On peut relever pour la nouvelle architecture modulaire les caractéristiques suivantes :
  - Structure du module de traitement (MP) multilinguale du message de façon pratiquement native.
  - Présence dans chaque sous-système d'une structure d'accueil pour option différente de traitement.
  - Possibilité de gestion allégée pour des réseaux ne nécessitant pas tous les outils de sécurité

- Réelle capacité d'administration à distance de façon sécurisée, y compris pour le maniement des clés. Cette nouvelle approche est confortée par l'introduction de l'option TCP en plus de UDP, TCP répondant mieux aux transferts volumineux répétés en géographique.

- La facilité de gestion des clés d'authentification et de cryptage (point qui avait condamné SNMPv2) En conservant l'impératif de clés différentes pour chaque équipement et pour chaque utilisateur, le mécanisme de « clés localisées » (expliqué plus loin) est transparent à l'utilisateur qui n'a qu'un password à manipuler.

- Approfondissement de la fonction *proxy-forwarder*, très importante dans la situation actuelle, certains équipements ne supportant qu'une version de SNMP (équipement ou gestionnaire). Une traduction d'une version SNMP dans une autre version s'impose dans ce cas.
- Sophistication des contrôles d'accès à la MIB de l'agent (VACM), selon de multiples critères :
  - Couple (utilisateur, modèle de sécurité)
  - sous-ensemble de la MIB (toute l'arborescence fille d'un point de l'espace)
  - options d'inclusion ou d'exclusion du sous-ensemble de la MIB

- notion de contextes différents dans un même équipement (avec identifiants différents)

• Capacité d'appliquer sur un agent SNMPv3 les fonctionnalités VACM, pour des commandes SNMPv1/v2c

• Richesse des documents qui détaillent avec une grande rigueur les fonctionnalités de la nouvelle architecture .

2.5 Principales nouveautés techniques propres à V3

1) Notion de **Principal** : entité au bénéfice de laquelle se déroule l'opération (utilisateur, réseau…)
2) SnmpEngineID : identifiant propre à l'équipement. La structure de cet identifiant peut être très variée[1], mais le plus souvent il est construit à partir du code constructeur d'une part et de l'adresse IP ou encore MAC (cisco préfère l'adresse MAC).
3) La communication entre les différents sous systèmes s'effectue sous forme de primitives (ASI cités plus haut) dont le mécanisme est introduit dans la RFC 2571 généraliste sur SNMPv3, et précisé dans les différentes MIBs spécialisées. Le livre de Stallings [10] explique bien ces mécanismes
4) Utilisation d'une nouvelle PDU (REPORT) réservée exclusivement à la communication entre machines SNMP. Cette PDU introduite par SNMPv2 n'avait pas pu être utilisée.
5) Les nouvelles fonctionnalités des tables, introduite par le SMIv2 sont pleinement utilisées (création/suppression dynamique de rangées dans les tables, indices pouvant ne pas appartenir à la table, indices rendus non-accessibles…).
6) Le mécanisme de génération des clés d'authentification et de cryptage était déjà introduite dans les pre-versions V3 concurrentes (SNMPv2U, SNMPv2*). La moulinette (Hash function) qui génère une clé « localisée» (par équipement) à partir de l'identifiant snmpEngineID, et du password (par utilisateur) est décrite dans Stallings [10] et dans la RFC2574.
7) La notion de temps « authoritative », c'est à dire qui prévaut. C'est le Command Responder (agent) qui détient le temps. Ce temps est contenu dans deux objets (nombre de redémarrages de l'équipement, temps écoulé depuis le dernier redémarrage). On n'a pas retenu le maintien (trop délicat) d'une horloge absolue. Lors de l'accès de l'agent, le temps figurant dans le message ne doit pas différer de +/-150s de celui de l' agent (nombre de Boots étant en accord). Sinon le message est rejeté comme non authentique..
8) Création d'utilisateurs : le processus de création des utilisateurs sur l'agent rappelle le mécanisme dans un système d'exploitation. On procédera au clonage d'un utilisateur existant (clonage en général limité à une opération). L'utilisateur doit ensuite changer son password (ce qui changera sa ou ses clés de sécurité sur cet équipement).
9) Le changement de clé peut se faire **à distance, sans cryptage** en toute sécurité (cette condition est nécessaire du fait que l'utilisation cryptage n'est pas une obligation dans SNMPv3). Il est toutefois conseillé de crypter si possible cet échange. Le mécanisme astucieux est décrit dans RFC2574 page 34-35.

2.6 Points faibles du modèle SNMPv3

- La modularité de l'architecture rend l'ensemble plus difficile à comprendre
- Chaque sous système gère ses tables et on trouvera de ce fait le même objet sous des noms différents (par exemple snmpEngineID, contextEngineID, msgAuthoritativeEngineID).
- La latitude future laissée aux développeurs d'ajouter des options de traitement (mention other dans les sous (systèmes) ouvre la porte à une dérive qui peut menacer à terme la compatibilité des produits. Dans un autre domaine (X25) la multiplicité des options avait mené à des implémentations nationales incompatibles de fait entre elles.
- Malgré un travail en profondeur considérable, SNMPv3 demeure un modèle scalaire, et reste à l'écart des techniques objet qui répondent mieux aux contextes actuels de développement.
- L'identifiant de la machine SNMP (snmpEngineID) peut être construit de multiples façon. Les risques de « duplicate » sont réels et beaucoup pensent qu'il fallait exclure l'utilisation de l'adresse IP dans la construction de cet identifiant.
- Peut être le plus important : les grands acteurs de l'administration réseau ne sont pas encore convaincus d'investir dans SNMPv3.

2.6 Structure du message

L'action des différents sous systèmes se traduit dans la composition du message (figures 4 et 6). La structure du message se divise en 4 champs :

1) version du message, permettant au Dispacher d'orienter le message vers la bonne version de traitement (la capacité multilinguale conseillée pour les entités SNMPv3 actuellement)

2) msgGlobalData, produisant les informations de service qui alimentent le sous système de traitement du message (MP), entre autres :
msgMaxsize : taille maximum du message supporté par l'expéditeur du message.
msgFlags : masque indiquant le degré de sécurisation du message, ainsi que la possibilité ou non de provoquer une PDU REPORT en retour (PDU de service entre machines SNMP)
msgSecurityModel : actuellement ne peut être que 3 (USM) dans un tel message.

3) MsgsecurityParameters
On retrouve une partie des objets présents dans la table USM. (identifiant Engine, paramètres de temps de la cible, Principal, et selon le degré de sécurité, le résultat de l'application des clés sur le message (Salt values).

4) ScopedPDU
La PDU est accompagnée (dans le cas général) de son nom de contexte et de l'identifiant correspondant. Pour illustrer le cas de contextes différents, un switch ATM par exemple, pourrait affecter à chaque interface un contexte différent, les adresses IP et MAC étant différentes. A ne pas confondre avec snmpEngineID correspondant à l'adresse sous laquelle cet équipement répond.

2.7 Déroulement des opérations

Le message donné en exemple est le résultat de l'opération, la réponse du Responder à l'Initiator. Pour parvenir à communiquer avec le Responder, cela suppose que l'expéditeur (l'Initiator) connaisse du destinataire (Responder), outre les clés requises, les renseignements suivants :

L'identifiant snmpEngineID cible
Les paramètres de temps de la cible (Responder) snmpEngineBoots et snmpEngineTime

Dans le cas étudié, si le processus de découverte du gestionnaitre (Initiator) n'a pas enregisté la machine authoritative, il doit y avoir une première requête avec les champs de sécurité vides. Le responder répond par une PDU REPORT (service entre machines SNMP), qui contient les paramètres désirés. Un deuxième get-request avec un message complet permettra le succès de l'opération.

Exemple de commandes SNMP mode ligne avec NET-SNMP (ancien UCD-SNMP) sur LINUX

Exemple de commande SNMPv1

```
snmpget -v 1 localhost public .1.3.6.1.2.1.1.4.0     (communauté = « public »)
   1.3.6.1.2.1.1.4.0 (sysContact) = M.Dupont

une commande SNMPv2C aurait la forme snmpget -v 2C ….
```

Exemple de commande SNMPv3 (exemple étudié plus loin)

```
Snmpget –v 3 –l autNoPriv –u xavier –a MD5 –A passsnmpv3 localhost .1.3.6.1.2.1.1.3.0
       1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
       Value: Timeticks: (249447) 0:41:34.47
```

Dans le mode ligne de NET-SNMP sur LINUX, tous les paramètres sont donnés au vol

Le niveau de priorité :	autNoPriv : (authentification mais pas de cryptage)
L'utilisateur *(Principal)*	xavier
Le protocole d'authentification	MD5
Le password de xavier	passsnmpv3 (lequel mouliné avec EngineID => clé d'authentification)
La cible (ici localhost)	
L'OID de la variable cible	1.3.6.1.2.1.3.0 : sysUpTime du groupe système, (.0) pour instance unique

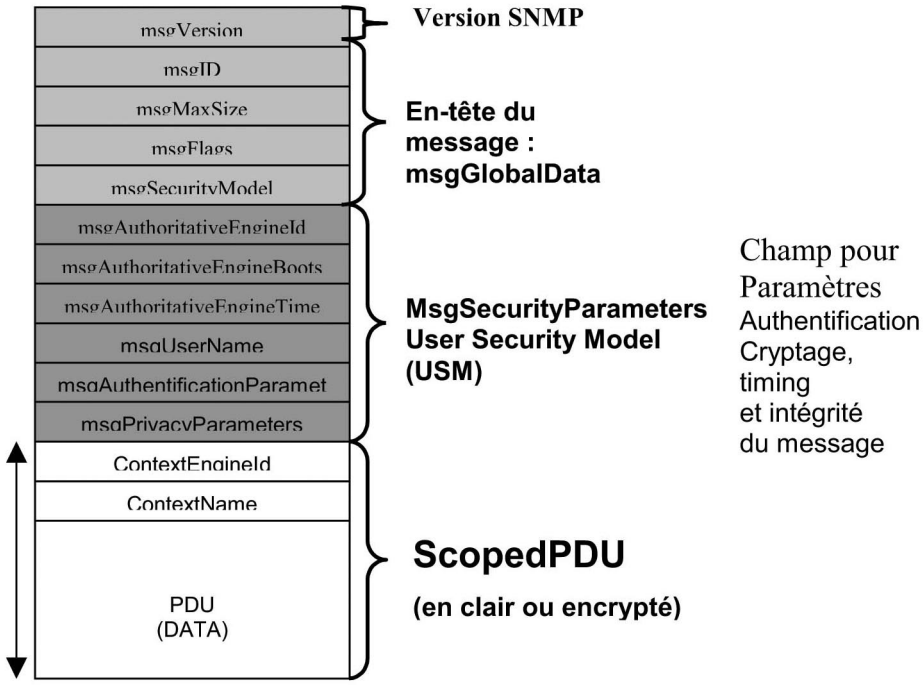


Figure 5

```
. Frame 36 (167 on wire, 167 captured)
Simple Network Management Protocol
    Version: 3

    Message Global Header
        Message Global Header Length: 13
        Message ID: 9
        Message Max Size: 1472
        Flags: 0x01
            .... .0.. = Reportable: Not set
            .... ..0. = Encrypted: Not set
            .... ...1 = Authenticated: Set
        Message Security Model: USM

    Message Security Parameters
        Message Security Parameters Length: 46
        Authoritative Engine ID:
         800007E501869E619D
        Engine Boots: 5
        Engine Time: 2494
        User Name: xavier
        Authentication Parameter:
         DA4D094554A2946557EB7F74
         Privacy Parameter:

    Context Engine ID: 800007E501869E619D
    Context Name:
    PDU type: RESPONSE
    Request Id: 0x8
    Error Status: NO ERROR
    Error Index: 0
    Object identifier 1:
       1.3.6.1.2.1.1.3.0 (SNMPv2-MIB::sysUpTime.0)
    Value: Timeticks: (249447) 0:41:34.47
```

Entête du message
Maximum message size : 1472
AuthNoPriv
Authentification mais pas cryptage
Reportable flag : 0 => aucune PDU
Report n'est à attendre en retour
Securitv Model : 3 = USM

Paramètres de sécurité
SnmpEngineID = 800007E501869E619D
Engine Boots et Time doivent coïncider avec ceux de la cible (Time à +/-150s près)
Auth params : l'application de la clé à l'arrivée doit donner le même résultat
Priv param : permet, si il y a cryptage , de retouver, avec la clé, le vecteur d'initialisation (IV) et ainsi de décrypter scoped PDU

Scoped PDU :
Champ susceptible d'être crypté si flag Contient l'option « encrypted ».
L'accès à la MIB est contrôlée par VACM Qui met en œuvre le mécanisme décrit plus bas (& VACM). 4 tables contiennent l'ensemble des paramètres qui déterminent ce contrôle.

2.8 Avancée fondamentale : la sécurisation des échanges en deux phases

La sécurisation dans SNMPv3 est réalisée à deux niveaux :

- Niveau du message : pris en charge par le sous système de sécurité , actuellement USM pour SNMPv3 Community Security Model pour SNMPv1 ou v2C.

  Authentification de la source, contrôle de l'intégrité du message

  Eventuellement cryptage des informations à protéger (on notera qu'il est possible de travailler sans authentification ni cryptage, mais pas sans contrôle de la fenêtre en temps).

  Contrôle du timing pour éviter les manipulations de ré-émission ou modification de séquence (maximum de +/- 150s d'écart pour l'accès à la machine « authoritative », si le nombre de redémarrage coïncide). L'accès à la machine « non authoritative » est moins sévère, le temps du message entrant ne devant pas retarder de plus de 150s.

- Niveau du contrôle d'accès aux objets de la MIB, lorsque le niveau précédent (authentification, intégrité du message, contrôle de la fenêtre en temps, éventuellement cryptage) est validé

VACM est actuellement le seul mécanisme de contrôle d'accès retenu dans SNMPv3. Rappelons que le contrôle d'accès sur la MIB d'un équipement, supportant SNMPv3, sera applicable à des commandes SNMPv1 ou v2C. C'est un des points remarquables de la nouvelle architecture. Ce deuxième niveau de sécurité est réglementé par les objets des différentes tables du sous-système VACM dont les fonctionnalités sont données ci dessous. Le diagramme d'état figure 7 explicite le rôle des objets impliqués.

SecurityToGroupTable

La première vérification est réalisée au niveau du couple (securityName, securityModel) , le Principal qui essaye d'accéder doit être associé à un modèle de sécurité, et appartenir à un groupe figurant dans la table « securityToGroupTable ». Il est important de voir qu'on peut trouver des groupes dont le modèle de sécurité est SNMPv1, ou v2C. Un couple (securityName, securityModel) ne peut appartenir qu'à un groupe, mais un Principal (securityName) peut appartenir à plusieurs groupes avec des modèles de sécurité différents.

Le modèle de sécurité v1 ou v2C, avec securityName associé est dérivé de la communauté (voir MIB SNMP-COMMUNITY-MIB dans la RFC 2576 traitant de la coexistence entre versions de SNMP).

VacmContextTable

On peut être amené à distinguer dans un équipement des zones à traiter séparément les unes des autres (on a évoqué des interfaces de commutateurs ATM par exemple, considérés comme des contextes différents). Cette table contiendra les noms des contextes.

VacmAccessTable

Cette table détermine la nature de l'accès demandé (Read/Write/Notify). Quatre index (groupe, modèle de sécurité, niveau de sécurité, et nom de contexte) déterminent la nature de l'opération.

VacmViewTreeFamilyTable

Cette table va conditionner la granularité plus ou moins précise de l'accès . l'OID SubTree (toutes les instances filles d'un point de l'espace de nommage), pourra être inclus ou exclu de l'accès (included/excluded) avec un raffinement procuré par un masque qui précisera sur l'OID de subTree, les instances précises que l'on veut inclure/exclure du champ d'accès.

Les instances qui peuplent ces tables conditionnent le résultat de la primitive isAccessAllowed, résultat qui autorise ou non finalement l'opération sur l'instance de l'objet cible.

Diagramme D'état de la primitive d'accès au sous-système de Contrôle d'Accès

Status= isAccessAllowed (securityModel, securityName, securityLevel, viewType, contextName, variableName)

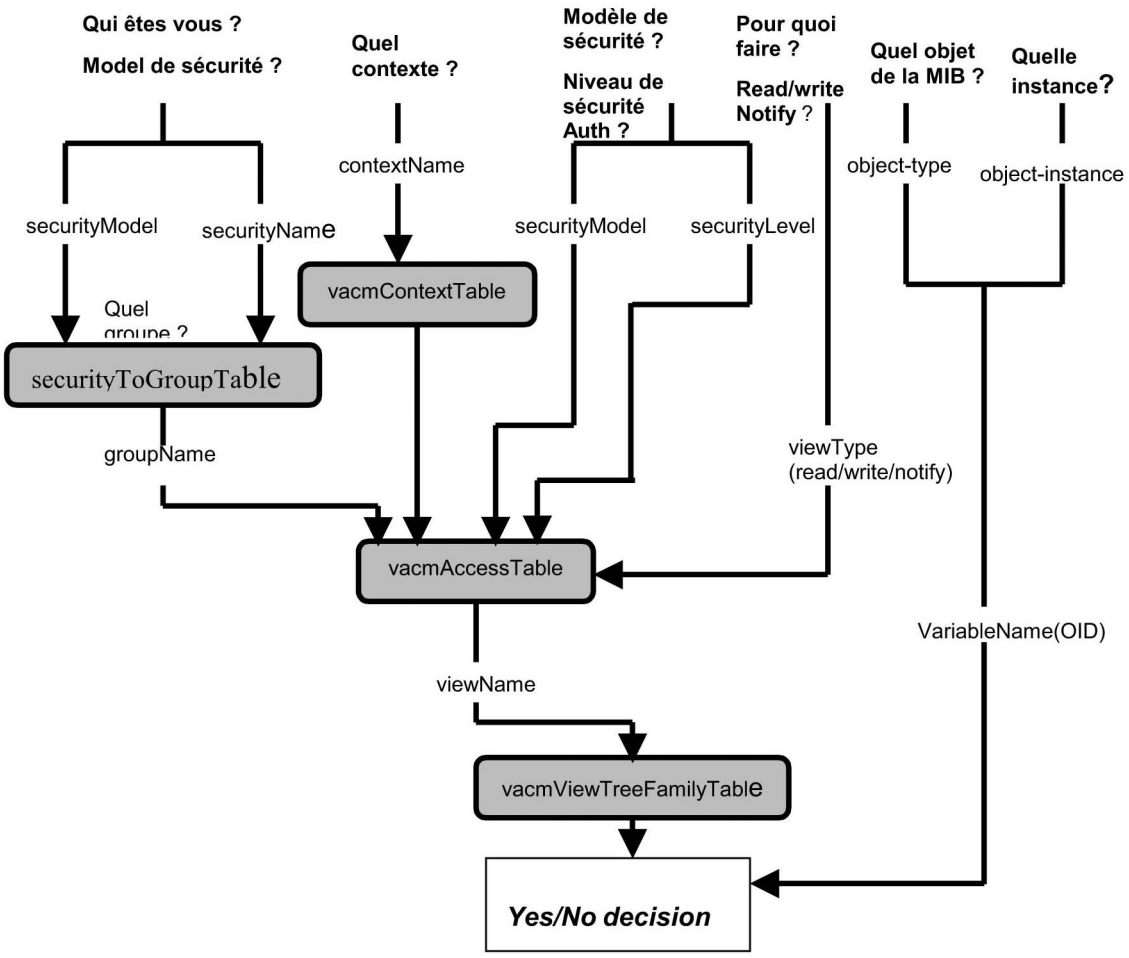


Figure 7

3. Coexistence entre les versions SNMPv1, V2c et V 3:

L'importance du parc installé SNMPv1 et v2c imposait que le nouveau protocole prenne en compte l'existant, d'une façon aussi transparente que possible pour l'utilisateur. S'il est assez aisé d'équiper les gestionnaires des moyens de communiquer avec les différentes versions de SNMP, le problème des agents ne supportant qu'une version de SNMP nécessitait d'être traité avec soin.

Cette situation avait déjà été rencontrée avec la coexistence entre version V1 et V2c.
Des modules MIB écrits selon le SMIv1 peuvent continuer à être accédés par des versions de protocole utilisant des PDU SNMPv2. Toutefois , si on veut être en conformité avec le SMIv2, un certain nombre de changements (environ 20) doivent être apportés dans la modification de la MIB écrite en SMIv1[6].

Deux autres types de situations de coexistence sont à considérer :

1) Les entités en présence sont les suivantes :
Gestionnaires ou agents ne parlant que V1 ou V1 et V2C

2) Gestionnaires et agents supportant V3 et en pratique capables de communiquer en V1 et V2C
Les mécanismes pour communiquer avec des entités de versions SNMP différentes sont de deux sortes :
- Capacité multilinguale
- Application proxy-forwarder à coté de l'application principale (Initiator, responder …).

La fonction multilinguale est par exemple ce qu'on doit trouver dans le sous système MP (Message Processing) dans les entités SNMP implémentant la nouvelle architecture, coté gestionnaire et coté agent (MP pour V1, V2C, V3).

Le seul réel problème est l'interrogation par une entité gestionnaire SNMPv1 d'une variable 64 bit provenant d'un agent écrit selon les règles du SMIv2. La traduction de la fonction proxy en SNMPv1doit ignorer ce type de variable.

Une autre situation est celle qui permet d'utiliser les fonctionnalités de VACM, (limitation de l'accès à la MIB de l'agent) dans le cas d'une requête provenant d'un gestionnaire V1 ou V2c. Une MIB (snmpCommunityMIB) décrit les objets qui sont utilisés, permettant la translation
communauté ⇦⇨ (securityName, securityModel).

On a ainsi défini des valeurs de **securityModel** pour V1(=1) et V2C(=2) ainsi que **messageprocessingModel** pour V1 (=0) et V2C(=1). Un mécanisme « Community-based Security Model » va être invoqué en plus des modèles de traitement V1/V2C présents dans le sous système MP de la nouvelle architecture.

Il faut souligner que ces mécanismes, fonctionnant dans les deux sens, sont invoqués par la fonction proxy-forwarder. Bien entendu un passage de V1/V2C à V3 ne peut donner que noAuth/noPriv comme niveau de sécurité, une translation dans l'autre sens perdant le niveau de sécurité comportant authentification ou cryptage.

4. Position des constructeurs vis à vis de SNMPv3

Les constructeurs ont été partagés entre les deux versions concurrentes au milieu des années 90. Aujourd'hui force est de constater que leur attitude varie entre l'attentisme, quelques implémentations d'évaluation (cisco), ou encore la superbe ignorance du nouveau protocole.

On peut dire que depuis deux années, les spécifications sont suffisamment précises pour implémenter SNMPv3 dans les gestionnaires et dans les agents.

Pour les gestionnaires, de petits constructeurs ont fait l'effort (SNMPC de Castlerock, MG-SOFT… et d'autres). Ajoutons aussi le remarquable travail de NET-SNMP (ancien UCD produit libre offrant agent et gestionnaire SNMPv1, v2C, v3 en mode ligne sur la plupart des plates-formes UNIX et Windows) .

- **Position de HP :**
  HP porte une lourde responsabilité dans l'attentisme de la communauté SNMP. La plate-forme HPOV est la référence dans le monde de l'administration réseau. Le retard de HP à intégrer en natif le nouveau protocole a été interprété que comme une marque de défiance vis à vis de SNMPv3. La solution (un peu confidentielle) de HPOV pour travailler en SNMPv3 est d'acquérir le logiciel commercial de SNMP RESEARCH ce qui n'est pas ce qu'on pourrait attendre d'un grand constructeur comme HP.

  La conséquence est que l'ensemble des matériels HP ignore SNMPv3 et privilégie à l'évidence les techniques WEB.

- **Position de CISCO**
  Cisco a introduit SNMPv3 partiellement sur certaines lignes de matériels (36XX , et 7000). Un aperçu est donné dans l'exemple qui suit, les fonctionnalités étant assez correctement vérifiées.

  Les autres matériels CISCO (switch 35XX ou 29XX) acceptent les instructions de configuration V3 mais redémarrent quand on applique une commande V3.

  En l'absence d'informations de réelles implémentations SNMPv3 dans des équipements réseaux, on se limitera à donner en annexe un apercu de l'exploration d'un serveur PPP 3640 de Cisco, accédé par un gestionnaire SNMPv3 UCD-SNMP sur LINUX. Les résultats avec un gestionnaire MG-SOFT (graphique) donnent les mêmes résultats

4.1 Exemple de fonctionnement d'équipements SNMPv3

Ce développement a été pris en charge par un stagiaire d'école d'ingénieur [7] au LAPP en 2000 sur un serveur PPP CISCO, (IOS 12.X) supportant SNMPv3. C'est un des rares équipements que nous avons pu tester assez loin avec SNMPv3. La version complète de l'exploration SNMPv3 avec l'ensemble des programmes PERL est donnée en référence [18]. Le traitement des TRAP RNIS (NUMERIS) permettrait de comptabiliser les temps de connexion si cela était souhaité. L'analyse des TRAP (v2C) nous a permis pendant quelques mois de vérifier périodiquement l'origine des appelants.

4.2 Evolution de SNMP

La retenue du marché à s'engager franchement dans le déploiement de SNMPv3 ont incité les parties prenantes à réfléchir sur les orientations futures de l'administration des réseaux ;

Le travail de réflexion sur l'évolution de SNMP et de son environnement est mené dans deux groupes de travail, EOS (Evolution Of SNMP) et SMIng (ng pour New Generation):

SNMP est reconnu comme un mécanisme élégant pour échanger des messages bien formalisés entre entités. Cependant combien d'achats de plate-formes onéreuses, censées apporter des solutions, se résument finalement à une implémentation d'un mécanisme simple qui est d'envoyer des ping 24H/24, 7 jours/7 …. On espérait quelque chose de simple à installer, à utiliser et à comprendre (une solution finale en somme) et on s'aperçoit que l'investissement en temps est lourd pour matérialiser une administration digne de ce nom.

Le protocole SNMP est simple, mais la difficulté réside dans la compréhension du contenu des MIBs, spécialement des MIBs constructeurs. La voie qui pourrait apporter un plus est peut être de compléter cette structure de données difficile à appréhender par de l'intelligence préparée spécialement à son environnement. On pense à des multiples et très légers « plug-in », « add-on » , peu onéreux qui pourrait enrichir spécifiquement les plate-formes génériques.

Cette approche, qui rejoint les approches alternatives à SNMP, est envisageable pour deux raisons :

- L'intelligence embarquée dans les équipements est sans commune mesure avec la situation des débuts de SNMP
- Les réseaux ont dépassé l'époque ou on vivait dans l'attente de la panne à réparer. Aujourd'hui les réseaux sont robustes et prêts à accepter une distribution coopérative de l'intelligence.

4.3 SMIng (ng pour New Generation), groupe EOS (Evolution Of SNMP)

Le but au départ du groupe de réflexion SMIng était de s'attaquer aux imperfections du SMIv2 et devait aboutir à une proposition pour un SMI renouvelé (SMIng). Les évaluations ont été conduites dans

l'environnement d'un projet parrallèle « libsmi », en y adjoignant un pré-compilateur (parser) ainsi qu'un générateur de code. Les résultats d'une première étape ont été rassemblés dans une présentation [14].

Le groupe de travail SMIng travaillait alors sous l'égide de l'IRTF/NMRG (Network Management Research

Group, IRTF Internet Research Task Force affilié à IETF). Un certain nombre de documents ont été produits :

- Une grammaire formelle (ABNF, Augmented Backus-Naur Form, RFC 2234)
- Trois modules MIB (core MIB modules)

L'IETF s'est alors impliqué dans le processus en créant le groupe « SMIng », dont les futures spécifications doivent être regroupées dans 4 parties distinctes :

- Spécifications pour un langage de base dans SMIng (core language)
- Spécifications pour les modules de base dans SMIng (core modules)
- Extensions du langage pour application à SNMP
- Extensions de langage pour adapter aux spécifications COPS-PR

Plusieurs Drafts ont été produits

SNMP Extended Protocol MIB
SNMP Row Operations Extensions
SNMP Object Identifier Compression
Efficient Transfer of Bulk SNMP data
SNMP Bulk Data Transfer Extensions

Les réflexions de ces groupes de travail s'attaquent aux aspects critiqués de longue date dans le SMI, imparfaitement corrigés dans les versions successives du SMIv1 et SMIv2.
Outre les types oubliés au départ (64 bit), qui ont causé tant de tracas, on pense avoir besoin dans le futur peut

être du type FLOAT, et peut être d'autres types de données.

Le manque d'outils pour s'assurer de la rigueur d'écriture des MIBs demandent des manipulations répétées pour corriger la formulation de nombreuses MIBs, y compris celles en provenance de l'IETF. Une couche supplémentaire libsmi, munie d'API, chapeautant les pré-compilateurs (parsers SMIv1/v2, SMIng, GDBM…) est à l'étude pour rendre plus rigoureuse l'écriture des MIBs ;
Le nommage des variables par OID répétant le chemin complet est trop lourd. L'encodage BER aggrave ce défaut. Un nouvel encodage, PER « Packed Encoding Rules » [20], utilise un mécanisme différent du TLV de BER . PER privilégie un encodage basé sur le type de donnée pour générer une représentation plus compacte. Le Type de TLV n'est généré que si cela est nécessaire pour éviter une ambiguïté (cas d'utilisation de CHOICE : version ASN.1 de UNION). PER génère en outre la longueur en octet de la valeur uniquement lorsque la taille de l'objet varie. PER essaye néanmoins d'encoder de façon plus économique. Les exemples cités dans la référence mettent en évidence un gain considérable (rapport jusqu'à 5 :1).

Enfin une extension des mécanismes de transfert de masse des données est également en chantier, le GET-BULK ne répondant pas suffisamment aux exigences actuelles.

Le calendrier de travail devait se dérouler au cours de l'année 2001, avec décision en octobre de poursuivre ou d'arrêter le groupe de travail.

5. Solutions alternatives ou associées à SNMP : coexistence HTTP et SNMP

Les remarques faites pour l'évolution de SNMP (intelligence accrue dans les équipements, omniprésence du WEB) sont applicables aux approches alternatives ou complémentaires.
Ces approches seront simplement citées , avec références pour étude plus complète. La présentation diapositive privilégie les schémas et fonctionnalités de ces architectures.

5.0 CMIP/CMIS de OSI

L'administration OSI est antérieure à SNMP et les concepts de l'administration ont été posé par OSI. CMIP/CMIS a suivi le destin des produits OSI, qui se cantonnent dans des niches confidentielles. Les déploiements de CMIP/CMIS comptent quelques secteurs très particuliers (surveillance radar européenne de l'aviation civile, certains secteurs des telecoms…). Pour des institutions qui avaient un besoin impératif de sécurité, pour lesquelles le coût de l'installation n'était pas dissuasif, CMIP/CMIS constituait une réponse.

Les applications OSI satisfont au modèle OSI, et s'appuient indifféremment sur le mode non connecté de OSI ou sur le mode connecté. Cette démarche est possible si on accepte de constituer un ensemble isolé, relié par lignes spécialisées par exemple.

Aujourd'hui, l'arrivée de SNMPv3, le poids de TCP/IP doivent amener les utilisateurs de CMIP/CMIS à s'interroger sur la pérennité de leur choix.

5.1 MRTG (Multi Router Traffic Grapher)

On ne peut parler de SNMP et du WEB sans se référer à une des premières tentatives de marier SNMP et le WEB. Ecrit en langage PERL, MRTG tourne sur de nombreuses plates-formes UNIX (LINUX essentiellement). Le mécanisme est simple : lecture d'une variable à intervalles périodiques (5mn par défaut), génération de pages WEB contenant des images GIF, visualisation par un navigateur WEB de l'historique des valeurs prises par la variable mesurée.

Cette approche est à classer dans la rubrique des techniques de « Monitoring », SNMP effectuant des GET-REQUEST , c'est à dire des lectures de compteurs essentiellement. La surveillance par graphiques retraçant une historique s'applique à tout équipement (routeurs, stations, switches…)

Cette technique est souvent associée à iptraffic, permettant de suivre l'évolution d'une distribution de protocoles sur une interface de routeur par exemple.

5.2 EWBM (Embedded Web-Based Management)

Dans cette approche, un serveur WEB est implanté dans l'équipement qui possède une adresse HTTP. On parle alors d'un agent WEB lequel communique par HTTP avec une application d'administration. Un agent WEB peut être sophistiqué comparé à l'agent SNMP.

Un autre avantage EWBM est qu'il peut tirer avantage des outils portables pour écrire l'agent WEB (par exemple WEB agent et WEB serveur peuvent être écrits en JAVA.

On peut trouver des exemples d'implémentations commerciales d'agents WEB, mais elles sont basées sur des protocoles propriétaires. La plupart des constructeurs d'équipements réseaux ont une offre d'agent WEB dans leur Hubs et Switches. Ces agents WEB coexistent et communiquent le plus souvent avec l'agent SNMP. La communication avec l'application de gestion couplée au navigateur (Advance Stack Assistant chez Hewlett Packard) se fait par HTTP.

Pour les switches, cette méthode est moins coûteuse que la sonde RMON attachée à chaque port.

5.3 DMI (Desktop Management Interface)

DMI est un standard de l'industrie produit par le consortium DMTF (Desktop Management Task Force) créé en 1992). Le but de DMTF était de développer et maintenir des outils de gestion standard pour les PC.

DMI se situe entre les composants des architectures machine et les logiciels qui les gèrent. Des entités « component Agents », qui peuvent être du logiciel (scruteur de virus), ou matériel (carte réseau) dialoguent avec une application résidente « Desktop management » à travers DMI. DMI consulte une base de données au format « MIF » (Management Information Format comparable à la MIB de SNMP, utilisant aussi ASN.1) et communique avec les deux entités par le jeu d'API.

Le but est d'arriver à gérer toutes les plate-formes sur un réseau de façon centralisée, indépendamment du constructeur et du système d'exploitation (DMI 2.0). DMI est assez différent de SNMP ; toutefois une tentative pour intégrer DMI dans SNMP a été tentée mais sans succès.

Entre 1996-98, la mission de DMTF a été élargie pour intégrer les standard de gestion existants (SNMP, CMIP, DMI et HTTP) et promouvoir le projet WBEM (Web Based Entreprise Management).

L'implication de Microsoft dans ce projet pourrait lui assurer une place significative dans la gestion des réseaux de PC.

5.4 Approche WBEM (Web-Based Entreprise Management)

Née d'une initiative de constructeurs mi 97 (Cisco, Intel, Compaq, BMC, Microsoft …), cette approche définit une charte commune ayant pour objectif de trouver des solutions d'administration interopérables, mais en intégrant les techniques existantes. Cette solution intègre CIM (Common Information Model) de Microsoft
Cela implique les points suivants :

- Description commune des données, indépendante de la plate-forme
- Protocole standard de publication des données et d'accessibilité de ces données.

WBEM n'a pas l'objectif de remplacer les protocoles établis (SNMP, et aussi CMIP) mais est censé intégrer ces techniques. Le navigateur WEB constitue l'interface utilisateur.

L'architecture détaillée de WBEM peut être trouvée dans les documents cités en référence [11] et [15]

Avantages de cette approche :
Réduit la complexité des solutions d'administration, puisqu'elle offre, par un navigateur WEB, une visualisation complète des données d'administration.

5.5 Approche JMAPI (Java Management API), JMX (Java Management Extensions) , EJB (Entreprise Java Beans)

Il s'agit la aussi d'une initiative constructeurs (SUN étant l'acteur principal, avec CISCO, Novell, Bay Netwoks…et d'autres). Cette solution à base de classes et d'interface JAVA est résolument « objet ». On retrouve la représentation et la modélisation uniforme des données.
On trouvera une description de ces mécanismes en référence [17].

Avantages : JAVA est indépendant du système et évite la gestion d'une plate-forme onéreuse. Cette approche est sans doute promise à des développements, la concentration de l'intelligence dans les équipements étant une donnée riche de promesses.

Inconvénients : lenteur excluant la gestion de grands réseaux, manque de règles établies pour implanter ces technologies.

Conclusion :

A la question « concurrents ou complémentaires », Nous savons maintenant que pour les petits réseaux, la gestion directe de l'équipement par le WEB est suffisamment attrayante pour supplanter la plate-forme onéreuse comme outil principal. Pour les grands réseaux, SNMP reste et restera l'outil incontournable, même si l'interface utilisateur peut utiliser le WEB comme interface utilisateur (HP Openview peut être lancé à partir d'un navigateur WEB).

SNMPv3 doit encore s'imposer et surmonter la réticence de la communauté SNMP. Les utilisateurs resteront sur la réserve tant que les grands acteurs du monde des réseaux et telecoms n'auront pas franchement donné leur appuis au nouveau protocole. Les fonctionnalités sont riches et permettent d'administrer de façon sécurisé un réseau en géographique, ce qui était impossible jusqu'à présent. Donc SNMPv3 s'imposera car le besoin de sécurité est exigé dans le monde ouvert de l'Internet.

L'arrivée progressive des équipements (agents) compatibles SNMPv3 permettra de franchir une première étape : se familiariser avec les gestionnaires existants SNMPv1/v2C pour organiser des hiérarchies d'utilisateurs avec des droits d'accès différents sur la MIB de l'équipement. On peut espérer que les logiciels de gestion intégreront en natif le nouveau protocole, décision indispensable pour convaincre la communauté SNMP. Comme souvent dans le passé, les petits développeurs de plateformes de gestion, aujourd'hui prêts pour SNMPv3 entraîneront les grands noms (Openview Network Node Manager de HP en particulier).

Pour la gestion de parcs de machines, SNMP avec plate-forme de gestion est moins attrayant pour les administrateurs système habitués à leur techniques propres. Cependant les techniques prônées par DMTF, après l'échec relatif de SMS (System Management Server de Microsoft™) auront leur place confortée par Microsoft, couplée à un navigateur WEB.

Il reste à percevoir les grandes tendances qui se dessinent assez nettement aujourd'hui. Le modèle « PULL » initiative préférentielle du gestionnaire, mis en place par SNMP évoluera de plus en plus vers le modèle « PUSH » , initiative croissante de l 'équipement grâce à l'intelligence implantée. Les solutions en gestation aujourd'hui s'intégreront de plus en plus dans le WEB, qui devient l'espace de travail préférentiel.

L'extrême diversité des solutions alternatives à SNMP, riches de promesses mais encore non stabilisées, plaident en final pour SNMP, protocole bien accepté et enfin finalisé.

Références

[1] RFC 2571 An Architecture for describing SNMP management Framework
[2] RFC 2572 Message Processing and Dispaching for SNMP
[3] RFC 2573 SNMP Applications
[4] RFC 2574 User-based Security Model (USM) for version 3 of SNMP
[5] RFC 2575 View-based Access control Model (VACM) for SNMP
[6] RFC 2576 Coexistence between version 1, 2, and 3 of managt…Framework
[7] Xavier Fournet - Configuration d'un serveur PPP cisco 3640 dans l'environnement SNMPv3 - (stage Ecole d'Ingénieur INSA au LAPP- été 2000)
[8] 10th IFIP/IEEE International Workshop on Distributed systems, Zurich, Oct 11-13, 1999
[9] T.Marshall Rose (livre) The Open Book
[10] W.Stallings (livre) SNMP, SNMPv2, SNMPv3… (3rd Edition)
[11] Mani Subramanian (livre) Network Management
[12] David Perkins Understanding SNMP
[13] The Simple Times publications sur le WEB
[14] F.Strauss SMIng…, DSOM'99 ETH Zürich, 12.10.1999
[15] Winston Bumpus DMTF Comdex Network Management presentation, April 3 2001
[16] J.P. Martin-Flattin Gestion des réseaux IP basée sur le WEB (GRES'99 Montréal juin 99)
[17] Ylian Saint-Hilaire Mémoire de Maitrise - Montréal
[18] Site WEB du LAPP wwwlapp.in2p3.fr
[19] Jeff Case Evolution of SNMP - 50th IETF – Minneapolis, MN march 18, 2001
[20] Introduction to ASN.1 and the Packed Encoding Rules (http://www.w3.org/HTTP-NG/asn.1.html)

ANNEXE A

Le Langage ASN.1 et l'encodage BER (Basic Encoding Rules)

Par rapport à OSI, les types primitifs sont réduits (INTEGER, OCTET STRING, OID, NULL). On trouve ensuite les type construits, SEQUENCE (concaténation de primitifs), les type dérivés des primitifs ((restriction des valeurs permises au type primitif). Quelques additifs sont en annexe.

On donne ci dessous un exemple de la grammaire (ASN.1) utilisée pour créer les variables de la MIB standard. Le résultat de cette invocation (instruction OBJECT TYPE)crée un « objet » **sysContact** du groupe **system**: **sysContact** est fille de l'objet **system**, lui même fille de l'objet **mib-2** , etc…

Les attributs de cet objet sont les suivants :

SYNTAX : une variable a un type primitif, construit ou dérivé d'un type primitif
STATUS : son statut (mandatory = obligatoire) tout agent SNMP doit incorporer cet objet
ACCESS : les opérations possibles sur cet objet : RW (accès possible en lecture et en écriture)
DESCRIPTION : description textuelle de la fonction de la variable créée

: := son assignation dans l'espace de nommage : system 4 signifie que sysContact est la variable No 4 du groupe system (1.3.6.1.2.1.1), et doit s'écrire alors 1.3.6.1.2.1.1.4
Le même mécanisme est mis en œuvre pour la création de la variable No 5 (sysName) du groupe system

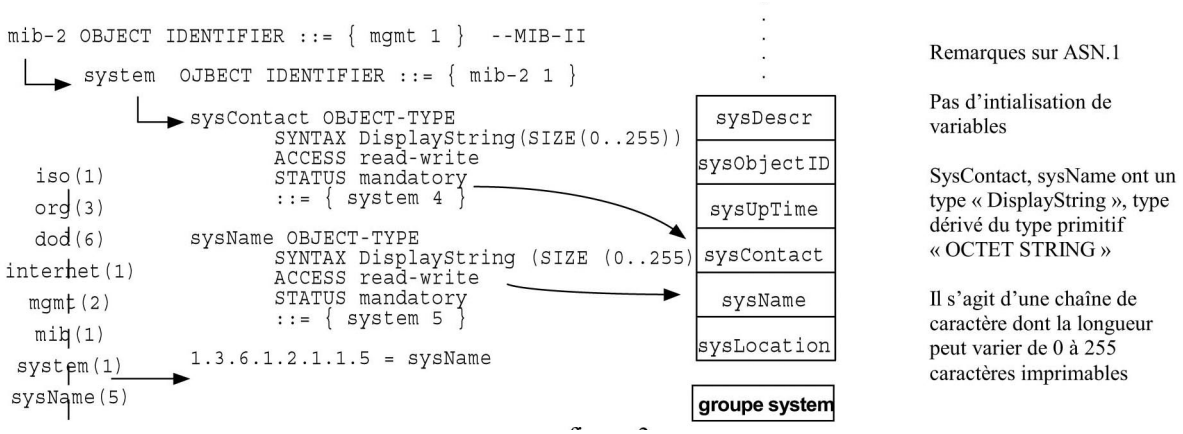


figure 3

On rappellera que la MIB standard (MIB-1 puis MIB-2) ne comportait au départ surtout des variables généralistes que tout équipement devait pouvoir intégrer. Par la suite des groupes de travail spécialisés ont développé des extensions au standard (MIB HUB, BRIDGE…) qui ont progressivement enrichi l'espace non constructeurs, lesquels avaient de moins en moins de raisons de poursuivre le développement de leurs MIBs privées

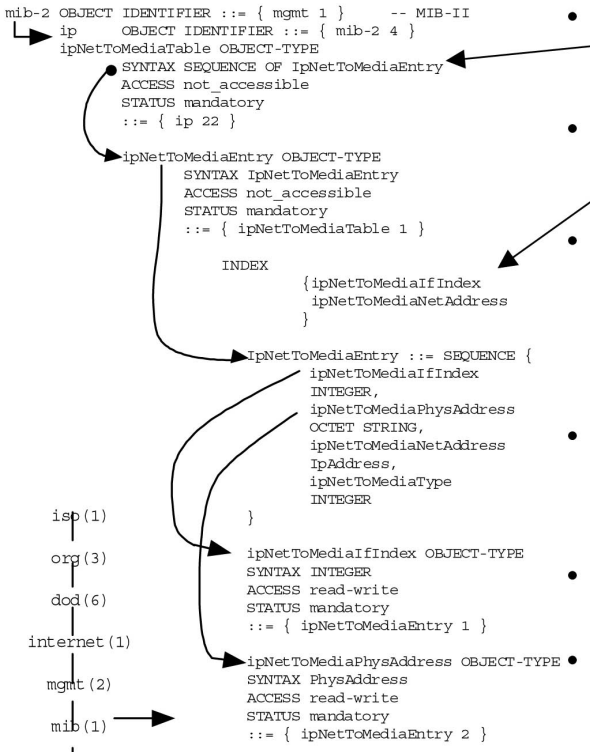
Accès aux variables :

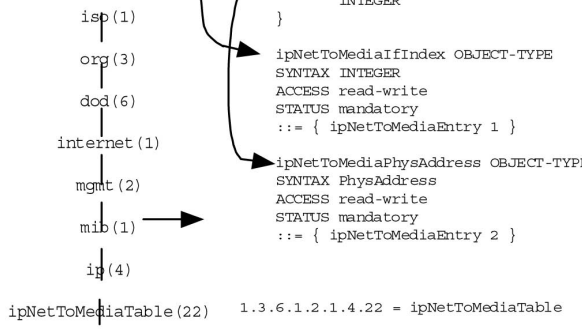
Le dialogue entre entités SNMP sur le réseau doit être indépendant des structures informatiques internes dans les équipements. C'est le rôle de l'encodage BER (Basic Encoding Rules), syntaxe concrète qui détaille la structure des données sur le réseau). C'est un mécanisme plus lourd que XDR (External Data Representation) , qui remplit la même fonction dans TCP/IP. On verra que c'est un des points que l'on tente de simplifier dans l'évolution de SNMP. . Toute variable circulant sur le réseau s'annonce de façon précise (TLV pour Type Longueur Valeur).
T dans TLV distingue 4 types de variables avec une étiquette dans T pour la variable précise.

Universal simples ou construit : (INTEGER, OCTET STRING …, SEQUENCE)
Application : (types connus dans S NMP seulement : IpAddress, Counter, Gauge …)
Context spécifique : (utilisé pour annoncer le début d'une PDU)
Private : prévu pour usage constructeur (en fait pas utilisé)

Exemple : la communauté « public » , dans le message SNMP on aura TLV : 04 06 7075626C6963
04 indique « Universal OCTET STRING » 06 annonce « 6 octets » 7075626C6963 annonce « public »

Commentaires sur les bizarreries de manipulation des tables :

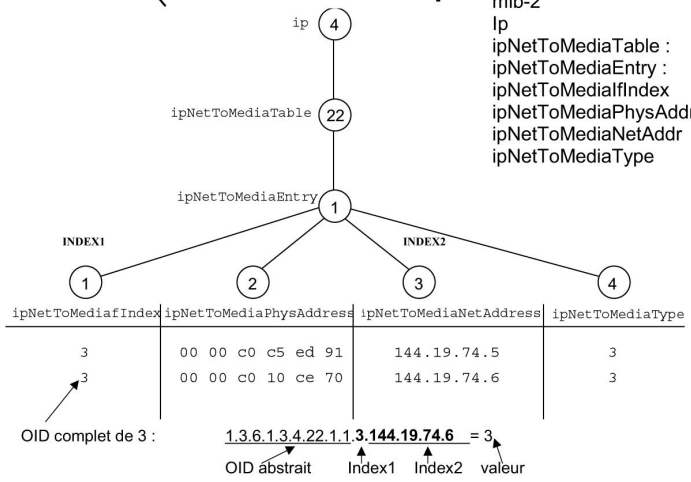




- En ASN.1 une table est un empilement ordonné « SEQENCE OF » de rangées (« SEQUENCE ») . L'équivalent de SEQUENCE en C est STRUCT (concaténation de variables simples).
- Dans une table ASN.1, les INDEX sont des **valeurs** de variables **appartenant** à la table, dont toutes les valeurs possibles conditionnent la profondeur de la table.
- Dans le cas étudié (le cache ARP), **ipNetToMediaIfIndex** est le No d'interface (plusieurs pour un routeur)
  i**pNetToMediaNetAddress** est l'adresse IP. On voit bien que le cache ARP complet du routeur sera obtenu en prenant toutes les valeurs des interfaces, puis toutes les valeurs des adresses IP .
- Dans un message échangé entre un gestionnaire et un agent, l'accès à une variable doit se faire en nommant la variable par son (OID) chemin complet, plus les INDEX (ici index1= 3, index2= 144.19.74.6)
- Dans SNMPv1, un INDEX peut être accédé en lecture (fonctionnalité inutile, et supprimé dans SMIv2). Dans les tables de l'architecture V3, les
- Index n'apparaissent pas à la lecture.

Dans l'espace de nommage (OID abstrait) :

mib-2	1.3.6.1.2.1.2	
Ip	1.3.6.1.2.1.2.4	
ipNetToMediaTable :	1.3.6.1.2.1.2.4.22	(non-accessible)
ipNetToMediaEntry :	1.3.6.1.2.1.2.4.22.1	(non-accessible)
ipNetToMediaIfIndex	1.3.6.1.2.1.2.4.22.1.1	RW
ipNetToMediaPhysAddr	1.3.6.1.2.1.2.4.22.1.2	RW
ipNetToMediaNetAddr	1.3.6.1.2.1.2.4.22.1.3	RW
ipNetToMediaType	1.3.6.1.2.1.2.4.22.1.4	RW



L'OID complet de 3 désigne une instance particulière (la case) de la table

A noter :
- La différence entre l'OID abstrait et l'OID complet quand on accède à une instance particulière de la table
- La lourdeur du nommage, bien plus long que la valeur elle même
- En SNMPv1 chaque case de table demande un échange (request-response) corrigé dans smiv2 avec getbulk
- L'inutilité de récupérer les valeurs d'index (corrigé dans SMIv2)

ANNEXE B Aperçu d'administration SNMPv3 sur un équipement cisco (serveur PPP)

Les tests complets figurent sur le serveur WEB du LAPP. Ce serveur PPP 3640 est le seul équipement sur lequel les fonctionnalités SNMPv3 ont pu être testés. Le résumé ci dessous donne un aperçu de la démarche familière de CISCO pour intégrer le protocole SNMP. En fait le terme de SNMPv3 n'apparaît pas vraiment, et la configuration SNMPv3 peut presque être réalisée en ignorant qu'il s'agit du nouveau protocole.

1. Configuration de la 3640 pour SNMPv3

1.1 Création d'un nouveau groupe

```
snmp-server group lapp v3 auth read v1default write v1default
```

crée un nouveau groupe "lapp" avec un accès SNMPv3 authentifié. La vue accessible en lecture et en écriture est v1default

1.2 Création d'un nouvel utilisateur

```
snmp-server user Xavier lapp v3 auth md5 abcdefgh
```

crée un nouvel utilisateur Xavier dans le groupe lapp avec une authentification par MD5. Le password est abcdefgh.

1.3 Visualisation des paramètres SNMP

On peut visualiser les paramètres relatif à SNMPv3 :

```
lapp-cisco#show snmp engineID
Local SNMP engineID: 00000009020000D0BAB69850
Remote Engine ID           IP-addr     Port

lapp-cisco#show snmp group
....
groupname: lapp                           security model:v3 auth
readview :v1default                       writeview: v1default
notifyview: <no notifyview specified>
row status: active
....

lapp-cisco#show snmp user
User name: Xavier
Engine ID: 00000009020000D0BAB69850
storage-type: nonvolatile         active
```

On remarque que l'engineID n'est pas un engineID version v3. On retrouve donc l'ancienne structure composée de 12 octets, les 4 premiers reprenant le code constructeur (9 pour CISCO) le reste étant spécifique à chaque constructeur. Ici CISCO à choisi de reprendre l'adresse Ethernet de la 3640.

1.4 Test de la configuration avec ucd-snmp

On peut donc tester cette configuration depuis une machine avec ucd-snmp :

```
[root@lappc-in14 mib3640]# snmpgetnext -v3 -l authNoPriv -u Xavier -a MD5 -A abcde-
fgh lapp-num system
system.sysDescr.0 = Cisco Internetwork Operating System Software IOS (tm) 3600
Software (C3640-IS-M), Experimental Version 12.0(19990819:200511) [rfh-120-
CSCdm27848t 129] Copyright (c) 1986-1999 by cisco Systems, Inc. Compiled Fri 20-
Aug-99 14:31 by rfh
```

En cas de mauvais password on obtient bien une erreur :

```
[root@lappc-in14 mib3640]# snmpgetnext -v3 -l authNoPriv -u Xavier -a MD5 -A
tototiti lapp-num system
snmpgetnext: Authentication failure
```

Glossaire :

ASN.1	Abstract Syntax Notation Nb 1 : langage de description utilisé pour écrire les MIB et le cadre d'administration (SMIv1 et SMIv2)
BER	Méthode d'encodage habituelle de l'ASN.1
CMIP	Common Management Information Protocol (administration réseau de OSI)
CMIS	Common Management Information Service
CMOT	CMIP Over TCP (tentative avortée d'utiliser CMIP/CMIS au dessus de TCP/IP
EOS	Evolution Of SNMP (groupe de travail pour la définition du futur de SNMP)
PDU	Protocol Data Unit (Message SNMP v1 : version, communauté, PDU (cad les données))
	SNMPv1 : Get-request, get-next-request, get-response, set-request, trap
	SNMPv2 : Get-request, get-next-request, getbulk, response, set-request, trap, inform, report
PER	Packed Encoding Rules (méthode d'encodage de ASN.1 plus compacte que BER)
MP	Message Processing: sous système de traitement du message dans SNMPv3 V1, V2C, V3 conseillés)
Principal	Entité au bénéfice de laquelle se déroule une opération SNMPv3
SMI	Structure of Management Information : cadre d'administration de SNMP
SMIv1	Cadre d'administration pour SNMPv1 (RFC1155 et 1212)
SMIv2	Cadre d'administration pour SNMPv2 repris presque inchangé pour SNMPv3
SMIng	Futur cadre d'administration pour SNMP à l'étude actuellement
SNMP	Simple Network Management protocol (lancé fin des années 90)
SNMPv2C	version de SNMP avec message de SNMPv1 mais utilisant SMIv2 (1995)
SNMPv2p	SNMPv2 party based (1993) version de SNMP qui a échoué pour le volet sécurité
SNMPv2U	version de SNMP sécurisée opposée à SNMPv2p
SNMPv2*	version de SNMP sécurisée concurrente de SNMPv2U
USM	User based security Model (modèle de sécurité pour SNMPv3 , basé sur password utilisateur)
VACM	View based Access Control Model (sous système de contrôle d'accès à la MIB de l'agent)