

Groupe de travail Réseau  
**Request for Comments : 2895**  
 RFC rendue obsolète : 2074  
 Catégorie : En cours de normalisation  
 Traduction Claude Brière de L'Isle

A. Bierman, Cisco Systems, Inc.  
 C. Bucci, Cisco Systems, Inc.  
 R. Iddon, 3Com, Inc.  
 août 2000

## Référence d'identifiant de protocole de MIB de surveillance de réseau à distance

### Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions d'amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Copyright

Copyright (C) The Internet Society (2000). Tous droits réservés.

### Résumé

Le présent mémoire définit une notation qui décrit les couches de protocole dans une encapsulation de protocole, spécifiquement pour l'utilisation au codage des valeurs INDEX pour le tableau protocolDirTable, qui se trouve dans la base de données d'informations de gestion de surveillance de réseau à distance (MIB RMON-2, *Remote Network Monitoring Management Information Base*) [RFC2021]. Les définitions des identifiants de couche de base de répertoire de protocole standard sont aussi incluses.

La première version du document des identifiants de protocole de RMON [RFC2074] a été partagée en une portion de référence en cours de normalisation (le présent document) et un document d'information. Le document Macros d'identifiant de protocole RMON [RFC2896] contient maintenant la portion non normative de cette spécification.

Le présent document rend obsolète la RFC2074.

## Table des Matières

1 Cadre de gestion de réseau SNMP.....	2
2. Vue d'ensemble.....	2
2.1 Termes.....	2
2.2 Relations avec la MIB de surveillance de réseau à distance.....	4
2.3 Relations avec le document Macros d'identifiant de protocole RMON.....	4
2.4 Relations avec la MIB ATM-RMON.....	4
2.5 Relations avec les autres MIB.....	5
3. Codage d'identifiant de protocole.....	5
3.1 Exemples de format d'INDEX de ProtocolDirTable.....	7
3.2 Format de macro d'identifiant de protocole.....	7
3.3 Évaluation d'un indice du tableau ProtocolDir.....	12
4. Macros d'identifiant de protocole de couche de base.....	12
4.1 Codage d'identifiant de base.....	13
4.2 Identifiants de protocole de couche de base.....	14
4.3 Couches d'encapsulation.....	17
5. Propriété intellectuelle.....	19
6. Remerciements.....	19
7. Références.....	19
8. Considérations relatives à l'IANA.....	21
9. Considérations sur la sécurité.....	21
10. Adresse des auteurs.....	21
Appendice A : Changements depuis la RFC 2074.....	21
11. Déclaration complète de droits de reproduction.....	22

## 1 Cadre de gestion de réseau SNMP

Le cadre de gestion SNMP consiste présentement en cinq composants majeurs :

- o Une architecture globale, décrite dans la [RFC2571].
- o Des mécanismes pour décrire et désigner les objets et événements pour les besoins de la gestion. La première version de cette structure d'informations de gestion (SMI, *Structure of Management Information*) est appelée SMIV1 et est décrite dans le STD 16, [RFC1155], [RFC1212] et [RFC1215]. La seconde version, appelée SMIV2, est décrite dans le STD 58, [RFC2578], [RFC2579] et [RFC2580].
- o Les protocoles de messages pour transférer les informations de gestion. La première version du protocole de message SNMP est appelée SNMPv1 et est décrite dans le STD 15, [RFC1157]. Une seconde version du protocole de message SNMP, qui n'est pas un protocole en cours de normalisation de l'Internet, est appelée SNMPv2c et est décrite dans la [RFC1901] et la [RFC1906]. La troisième version du protocole de message est appelée SNMPv3 et est décrite dans les [RFC1906], [RFC2572] et [RFC2574].
- o Opérations du protocole pour accéder aux informations de gestion. Le premier ensemble d'opérations du protocole et les formats de PDU associés sont décrits dans le STD 15, [RFC1157]. Un second ensemble d'opérations du protocole et des formats de PDU associés est décrit dans la [RFC1905].
- o Un ensemble d'applications fondamentales décrites dans la [RFC2573] et le mécanisme de contrôle d'accès fondé sur la vue décrit dans la [RFC2575].

Une introduction plus détaillée du cadre de gestion SNMP actuel se trouve dans la [RFC2570].

On accède aux objets gérés via un magasin virtuel d'informations, appelé la base de données d'informations de gestion (MIB, *Management Information Base*). Les objets de la MIB sont définis en utilisant les mécanismes définis dans le SMI.

Le présent mémoire ne spécifie pas un module de MIB.

## 2. Vue d'ensemble

La MIB RMON-2 [RFC2021] utilise des chaînes d'octets formatées hiérarchiquement pour identifier globalement les encapsulations de protocole individuelles dans le tableau protocolDirTable.

Ce guide contient des algorithmes et l'ensemble d'autorité des macros d'identifiant de protocole de couche de base, à utiliser dans les valeurs INDEX dans le tableau protocolDirTable.

Ceci est la seconde révision du document, et elle est destinée à remplacer la première moitié du premier document Identifiant de protocole RMON-2 [RFC2074].

### 2.1 Termes

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119].

Plusieurs termes qui sont utilisés tout au long de ce document, ainsi que dans la MIB RMON-2 [RFC2021], doivent être présentés :

protocole parent :

Aussi appelé 'parent' ; c'est l'identifiant de protocole encapsulant pour une couche de protocole spécifique, par exemple, IP est le protocole parent de UDP. Noter que les couches de base ne peuvent pas avoir de protocoles parents. Ce terme peut être utilisé pour se référer à un protocole encapsulant spécifique, ou peut être utilisé de façon générique pour se référer à tout protocole encapsulant.

protocole fils :

Aussi appelé 'fils' ; c'est un identifiant de protocole encapsulé pour une couche de protocole spécifique. Par exemple, UDP est un protocole fils de IP. Ce terme peut être utilisé pour se référer à un protocole encapsulé spécifique, ou il peut être

utilisé de façon générique pour se référer à tout protocole encapsulé.

identifiant de couche :

Fragment de chaîne d'octets représentant une couche ou sous-couche d'encapsulation de protocole particulière. Un fragment consiste en exactement quatre octets, codés dans l'ordre des octets du réseau. Si ils sont présents, les identifiants de couches filles pour un protocole DOIVENT avoir des valeurs uniques les unes par rapport aux autres. (Voir les détails au paragraphe 3.3.)

protocole :

Une couche de protocole particulière, comme spécifié par les règles de codage de ce document. Se réfère usuellement à une seule couche dans une certaine encapsulation. Noter que ce terme est parfois utilisé dans la MIB RMON-2 [RFC2021] pour désigner une chaîne pleinement spécifiée d'identifiant de protocole. Dans ce cas, la chaîne d'identifiant de protocole est désignée pour sa couche supérieure. Un protocole désigné peut aussi se référer à toute encapsulation de ce protocole.

chaîne d'identifiant de protocole :

Chaîne d'octets représentant une encapsulation de protocole particulière, comme spécifié par les règles de codage du présent document. Cette chaîne est identifiée dans la MIB RMON-2 [RFC2021] comme l'objet protocolDirID. Une chaîne d'identifiant de protocole est composée d'un ou plusieurs identifiants de couche lus de gauche à droite. L'identifiant de couche de gauche spécifie une encapsulation de couche de base. Chaque identifiant de couche à droite spécifie une encapsulation de protocole de couche fille.

macro d'identifiant de protocole : aussi appelée macro PI (PI, *Protocol Identifier*) ; c'est une construction textuelle comme une macro utilisée pour décrire un protocole de réseautage particulier. Seuls les attributs de protocole qui sont importants pour l'utilisation de RMON sont présentés. Noter que le terme 'macro' est historique, et que les macros PI ne sont pas de réelles macros, ni ne sont des macros ASN.1. L'ensemble actuel des macros PI RMON publiées se trouve dans le document "Macros d'identifiant de protocole RMON" [RFC2896].

La macro PI sert à plusieurs fins :

- Désigner le protocole à utiliser dans la MIB RMON-2 [RFC2021].
- Décrire comment le protocole est codé en une chaîne d'octets.
- Décrire comment les protocoles fils sont identifiés (si applicable) et codés en chaîne d'octets.
- Décrire quels protocolDirParameters sont permis pour le protocole.
- Décrire comment l'objet protocolDirType associé est codé pour le protocole.
- Fournir des références à la documentation d'autorité pour le protocole.

macro identifiant de variante de protocole :

Aussi appelé macro de variante PI ; sorte particulière de macro PI, utilisée pour décrire une certaine couche de protocole, qui ne peut pas être identifiée avec une structure déterministe, et (usuellement) hiérarchique, comme la plupart des protocoles d'inter réseautage.

Noter que la macro de variante PI et la macro PI sont définies avec un seul ensemble de règles syntaxiques (voir le paragraphe 3.2) sauf que des sous-classes différentes sont requises pour chaque type.

Un protocole identifié avec une macro de variante PI est en fait une variante d'une encapsulation bien connue qui peut être présente dans le tableau protocolDirTable. Cela est utilisé pour documenter les protocoles alloués par l'IANA, qui sont nécessaires pour identifier les protocoles qui ne peuvent en pratique être identifiés par l'examen d'un 'trafic réseau approprié' (par exemple, les paquets qui les portent). Tous les autres protocoles (qui peuvent être identifiés par examen du trafic réseau approprié) DEVRAIENT être documentés en utilisant la macro d'identifiant de protocole. (Voir les détails au paragraphe 3.2.)

paramètre de protocole :

Un seul octet, correspondant à un identifiant de couche spécifique dans l'identifiant de protocole. Cet octet est un gabarit binaire qui indique des fonctions ou capacités particulières que cet agent fournit pour le protocole correspondant. (Voir les détails au paragraphe 3.2.6.)

chaîne paramètres de protocole :

Chaîne d'octets, qui contient un paramètre de protocole pour chaque identifiant de couche dans l'identifiant de protocole. Cette chaîne est identifiée dans la MIB RMON-2 [RFC2021] comme l'objet protocolDirParameters. (Voir les détails au paragraphe 3.2.6.)

INDEX protocolDirTable :

Paire de chaînes d'octets Identifiant de protocole et paramètres de protocole qui ont été converties en une valeur d'INDEX, conformément aux règles de codage du paragraphe 7.7 de la [RFC1902].

pseudo protocole :

Convention ou algorithme utilisé seulement dans le présent document pour les besoins du codage des chaînes d'identifiant de protocole.

arborescence d'encapsulation de protocole :

Les encapsulations de protocole peuvent être organisées en un arbre inversé. Les nœuds de la racine sont les encapsulations de base. Les nœuds fils, s'il y en a, d'un nœud de l'arbre sont les encapsulations des protocoles fils.

## 2.2 Relations avec la MIB de surveillance de réseau à distance

Le présent document est destiné à identifier les règles de codage pour les objets CHAINE D'OCTETS protocolDirID et protocolDirParameters. Les tableaux RMON-2, comme ceux de la nouvelle distribution de protocole, les groupes Host, et Matrix, utilisent un INDEX ENTIER local (protocolDirLocalIndex) plutôt que des chaînes INDEX protocolDirTable complètes pour identifier les protocoles à des fins de comptage. Seul le protocolDirTable utilise le protocolDirID et les chaînes protocolDirParameters décrites dans le présent document.

Ce document est volontairement séparé de la MIB des objets RMON-2 [RFC2021] pour permettre une mise à jour de ce document sans nouvelle publication des objets de la MIB.

Le présent document n'expose pas l'auto découverte ni l'auto remplissage du protocolDirTable. Cette fonctionnalité n'est pas explicitement définie par la norme RMON. Un agent DEVRAIT remplir le répertoire avec les protocoles 'intéressants' desquels dépendent les applications envisagées.

## 2.3 Relations avec le document Macros d'identifiant de protocole RMON

Le document d'origine sur les identifiants de protocole RMON [RFC2074] contient le matériel de référence de répertoire de protocoles, ainsi que de nombreux exemples de macros d'identifiant de protocole.

Ces macros ont été déplacées dans un document distinct intitulé Macros d'identifiant de protocole RMON [RFC2896]. Cela va permettre que le texte normatif (le présent document) avance sur la voie de la normalisation avec la MIB RMON-2 [RFC2021], tandis que la collection de macros PI est conservée dans une RFC informative.

Le document Macros PI est volontairement séparé du présent document pour permettre la mise à jour de la liste des macros PI publiées sans republication des objets de la MIB ni des règles de codage. Les macros d'identifiant de protocole soumises par le groupe de travail RMON et la communauté au sens large (de la liste de diffusion du groupe de travail RMONMIB à 'rmonmib@ietf.org') seront collectées, examinées par le groupe de travail RMONMIB, et (si elles sont approuvées) ajoutées à une version ultérieure du document Macros PI.

Les soumissions de macros seront collectées dans les fichiers MIB de l'IANA sous le répertoire "ftp://ftp.isi.edu/mib/rmonmib/rmon2\_pi\_macros/" et dans le fichier d'archive de la liste de diffusion du groupe de travail RMONMIB [www.ietf.org/mail-archive/working-groups/rmonmib/current/maillist.htm](http://www.ietf.org/mail-archive/working-groups/rmonmib/current/maillist.htm) .

## 2.4 Relations avec la MIB ATM-RMON

Le Forum ATM a normalisé les "Extensions de MIB de surveillance à distance pour les réseaux ATM" (MIB ATM-RMON) [AF-NM-TEST], qui fournit des statistiques, hôte, matrice et capacité matrixTopN de style RMON, pour le trafic de cellule NSAP fondée sur l'adresse (couche 5 d'adaptation ATM, AAL-5).

### 2.4.1 Agrégation d'accès

Il est possible de corréler les données de MIB ATM-RMON avec les collections de RMON-2 fondées sur le paquet [RFC2021], mais seulement si le 'portSelGrpTable' et le 'portSelTable' ATM-RMON sont configurés pour fournir le même niveau d'agrégation d'accès qu'utilisé dans la collection fondée sur le paquet. Cela exigera qu'un 'portSelectGroup' ATM-RMON contienne un seul accès, dans le cas de sources de données RMON traditionnelles.

### 2.4.2 Transpositions d'encapsulation

Le document RMON PI ne contient pas de prise en charge explicite de macro PI pour "l'encapsulation multi protocoles sur

la couche 5 d'adaptation ATM" [RFC1483], ou "l'émulation de LAN sur ATM" (LANE) de l'ATM Forum [AF-LANE]. À la place, une sonde doit 'faire coller' l'encapsulation ATM à une des couches de base définies dans le présent document (c'est-à-dire, llc, snap, ou vsnap) sans considération de la façon dont les données brutes sont obtenues par l'agent (par exemple, multiplexage de circuit virtuel ou multiplexage de commande de liaison logique (LLC, *Logical Link Control*) ou formats acheminés contre formats pontés). Voir au paragraphe 3.2 les détails de l'identification et du décodage d'une couche de base particulière.

Un système de gestion de réseau (NMS, *network management system*) peut déterminer certains des détails d'encapsulation omis en examinant le type d'interface (ifType) de la source de données pour une collection RMON particulière :

dataSource ifTypes de la RFC 1483 : - aal5(49)

dataSource ifTypes de LANE : - aflane8023(59)  
- aflane8025(60)

Ces sources de données exigent la mise en œuvre du tableau ifStackTable par la MIB des interfaces [RFC2233]. Il est possible que certaines mises en œuvre utilisent des valeurs de dataSource qui indiquent un ifType de 'atm(37)' (parce que le ifStackTable n'est pas pris en charge) cependant ceci est fortement déconseillé par le groupe de travail RMONMIB.

### 2.4.3 Comptage du trafic ATM dans les collections RMON-2

Les tableaux (host/matrix/topN) de couche application (AL, *Application Layer*) et de couche réseau (NL, *Network Layer*) RMON-2 exigent que les compteurs d'octets soient incrémentés de la taille de la trame concernée, et non de la taille de la trame attribuée à un protocole donné.

Les mises en œuvre de sondes doivent utiliser la taille de trame AAL-5 (et non la taille de charge utile AAL-5 ni la taille de trame MAC encapsulée) comme 'taille de trame' pour les besoins de l'incrémentation des compteurs d'octets RMON-2 (par exemple, 'nlHostInOctets', 'alHostOutOctets').

Le groupe de travail RMONMIB n'a pas traité les questions qui se rapportent à la capture de paquet du trafic fondé sur AAL-5. Donc, c'est une question spécifique de la mise en œuvre de décider si les octets de bourrage (c'est-à-dire, d'un circuit virtuel multiplexé de la RFC1483, du trafic 802.3 ou 802.5 ponté, ou du trafic LANE) sont représentés dans l'objet de MIB RMON-1 'captureBufferPacketData'. Normalement, le premier octet de la trame capturée est le premier octet de l'adresse MAC de destination (DA, *Destination Address*).

## 2.5 Relations avec les autres MIB

Le document Référence des identifiants de protocole RMON est destiné à être utilisé avec le tableau protocolDirTable au sein de la MIB RMON. Il n'est pertinent pour aucune autre MIB, ni destiné à être utilisé avec aucune autre MIB.

## 3. Codage d'identifiant de protocole

Le tableau protocolDirTable est indexé par deux CHAINES D'OCTETS, protocolDirID et protocolDirParameters. Pour coder l'indice du tableau, chaque chaîne de longueur variable est convertie en un fragment d'IDENTIFIANT D'OBJET, selon les règles de codage du paragraphe 7.7 de la [RFC1902]. Les fragments d'index sont alors simplement enchaînés. (Pour les détails, se référer aux figures 1a - 1d ci-dessous.)

La première CHAÎNE D'OCTETS (protocolDirID) se compose d'un ou plusieurs "identifiants de couche" de quatre octets. La chaîne entière identifie de façon univoque un certain nœud dans l'arborescence d'encapsulation de protocole. La seconde CHAÎNE D'OCTETS (protocolDirParameters) qui contient un nombre correspondant de paramètres spécifiques du protocole de un octet, un pour chaque identifiant de couche de quatre octets dans la première chaîne.

Une couche de protocole est normalement identifiée par une seule valeur de 32 bits. Chaque identifiant de couche est codé dans l'INDEX de CHAÎNE D'OCTETS ProtocolDirID comme quatre sous-composants [ a.b.c.d ], où 'a' à 'd' représentent chaque octet de la valeur de 32 bits dans l'ordre des octets du réseau. Si une couche de protocole particulière ne peut pas être codée sur 32 bits, elle doit alors être définie comme un protocole 'ianaAssigned' (voir ci-dessous les détails des protocoles alloués par l'IANA).

Les figures qui suivent montrent les différences entre les codages IDENTIFIANT D'OBJET et CHAÎNE D'OCTETS de la chaîne d'identifiant de protocole.

**Figure 1a : Format d'INDEX de protocolDirTable**

```

+---+-----+-----+-----+-----+
| c !               | c !               |
| n ! protocolDirID | n ! Parameters    |
| t !               | t ! protocolDir   |
+---+-----+-----+-----+

```

**Figure 1b : Format CHAINE D'OCTETS de protocolDirTable**

```

protocolDirID
+-----+
|               |
|               |
|               |
|               |
+-----+

```

4 \* N octets

```

protocolDirParameters
+-----+
|       |
| N octets |
|       |
+-----+

```

N est le nombre d'identifiants de couche de protocole requis pour l'encapsulation entière du protocole désigné. Noter que la couche suivant la couche de base identifie habituellement un protocole de couche réseau, mais ce n'est pas toujours le cas, (en particulier pour les enfants de la couche de base 'vsnap').

**Figure 1c : Exemple de format d'INDEX de protocolDirTable**

```

protocolDirID                protocolDirParameters
+---+-----+-----+-----+-----+-----+-----+-----+
| c | proto | proto | proto | proto | c | par | par | par | par |
| n | base  | L(B+1) | L(B+2) | L(B+3) | n | ba- | L3 | L4 | L5 |
| t | fanions | L3  | L4  | L5  | t | se  |   |   |   |
+---+-----+-----+-----+-----+-----+-----+
| 1 | 4     | 4   | 4   | 4   | 1 | 1  | 1  | 1  | 1  |
+---+-----+-----+-----+-----+-----+
compte
sous OID

```

Lorsque codée dans un INDEX de protocolDirTable, chacune des deux chaînes doit être précédée par un sous composant Longueur. Dans cet exemple, N égale '4', le premier champ 'cnt' contiendrait la valeur '16', et le second champ 'cnt' contiendrait la valeur '4'.

**Figure 1d : Exemple de format CHAINE D'OCTETS de protocolDirTable**

```

protocolDirID
+-----+-----+-----+-----+
| proto | proto | proto | proto |
| base  | L3    | L4    | L5    |
|       |       |       |       |
+-----+-----+-----+-----+
compte
| 4     | 4     | 4     | 4     |
+-----+-----+-----+-----+
d'octets

```

```

protocolDirParameters
+---+---+---+---+
| par | par | par | par |
| ba- | L3 | L4 | L5 |
| se  |   |   |   |
+---+---+---+---+
compte
| 1 | 1 | 1 | 1 |
+---+---+---+---+
d'octets

```

Bien que cet exemple indique quatre protocoles encapsulés, en pratique, tout nombre non zéro d'identifiants de couche peut être présent, théoriquement limité seulement par les restrictions de longueur d'IDENTIFIANT D'OBJET, comme spécifié au paragraphe 3.5 de la [RFC1902].

### 3.1 Exemples de format d'INDEX de ProtocolDirTable

Les fragments d'identifiant PI sont des exemples de quelques valeurs d'indices codés de protocolDirTable pour diverses encapsulations.

-- HTTP ; fragments comptés à partir de IP et au dessus de ether2.ip.tcp.www-http =  
16.0.0.0.1.0.0.8.0.0.0.6.0.0.0.80.4.0.1.0.0

-- SNMP sur UDP/IP sur SNAP  
snap.ip.udp.snmp = 16.0.0.0.3.0.0.8.0.0.0.17.0.0.0.161.4.0.0.0.0

-- SNMP sur IPX sur SNAP  
snap.ipx.snmp = 12.0.0.0.3.0.0.129.55.0.0.144.15.3.0.0.0

-- SNMP sur IPX sur raw8023  
ianaAssigned.ipxOverRaw8023.snmp = 12.0.0.0.5.0.0.0.1.0.0.144.15.3.0.0.0

-- IPX sur LLC  
llc.ipx = 8.0.0.0.2.0.0.0.224.2.0.0

-- SNMP sur UDP/IP sur toute couche de liaison  
ether2.ip.udp.snmp = 16.1.0.0.1.0.0.8.0.0.0.17.0.0.0.161.4.0.0.0.0

-- IP sur toute couche de liaison ; le codage de base est IP sur ether2  
ether2.ip = 8.1.0.0.1.0.0.8.0.2.0.0

-- AppleTalk phase 2 sur ether2  
ether2.atalk = 8.0.0.0.1.0.0.128.155.2.0.0

-- AppleTalk phase 2 sur vsnap  
vsnap.apple-oui.atalk = 12.0.0.0.4.0.8.0.7.0.0.128.155.3.0.0.0

### 3.2 Format de macro d'identifiant de protocole

L'exemple suivant est destiné à introduire la macro protocol-identifier. Cette construction de style macro est utilisée pour représenter à la fois les protocoles et les variantes de protocole.

Si le composant 'VariantOfPart' de la macro est présent, la macro représente alors une variante de protocole plutôt qu'un protocole. Cette clause n'est actuellement utilisée que pour les protocoles alloués par l'IANA, énumérés sous la couche de base 'ianaAssigned'. Le composant VariantOfPart DOIT être présent pour les protocoles alloués par l'IANA.

#### 3.2.1 Conventions lexicales

Le langage PI définit les mots-clés suivants :

ADDRESS-FORMAT (*format d'adresse*)  
 ATTRIBUTES (*attributs*)  
 CHILDREN (*enfants*)  
 DECODING (*décodage*)  
 DESCRIPTION  
 PARAMETERS  
 PROTOCOL-IDENTIFIER (*identifiant de protocole*)  
 REFERENCE  
 VARIANT-OF

Le langage PI définit les éléments de ponctuation suivants :

{ accolade gauche  
 } accolade droite  
 ( parenthèse gauche  
 ) parenthèse droite

, virgule  
 ::= deux fois deux points et signe égal  
 -- deux tirets

### 3.2.2 Notation des descriptions syntaxiques

Une forme étendue de la notation BNF est utilisée pour spécifier la syntaxe du langage PI. Les règles de cette notation sont données ci-dessous :

- \* Les valeurs littérales sont spécifiées entre guillemets, par exemple "REFERENCE"
- \* Les éléments non terminaux sont entourés des caractères inférieur à (<) et supérieur à (>) par exemple <parmList>
- \* Les éléments terminaux sont spécifiés sans guillemets ou caractères supérieur à et inférieur à, par exemple 'lcname'
- \* Une barre verticale (|) est utilisée pour indiquer un choix entre éléments, par exemple 'nombre | hstr'
- \* Des ellipses sont utilisées pour indiquer que l'élément précédent peut être répété une ou plusieurs fois, par exemple <parm>...
- \* Des crochets angulaires sont utilisés pour enclore des éléments facultatifs, par exemple [ ", <parm> ]
- \* Un caractère égal (=) est utilisé pour dire "défini comme," par exemple '<protoName> = pname'

### 3.2.3 Grammaire du langage PI

Ce qui suit sont les "terminaux" de la grammaire et sont identiques aux mêmes éléments lexicaux tirés du langage de module de MIB, sauf pour hstr et pname :

```

<lc> = "a" | "b" | "c" | ... | "z"
<uc> = "A" | "B" | "C" | ... | "Z"
<lettre> = <lc> | <uc>
<chiffre> = "0" | "1" | ... | "9"
<hdigit> = <chiffre> | "a" | "A" | "b" | "B" | ... | "f" | "F"
<lcname> = <lc> [ <lcrest> ]
<lcrest> = ( <lettre> | <chiffre> | "-" ) [ <lcrest> ]
<pname> = ( <lettre> | <chiffre> ) [ <pnrest> ]
<pnrest> = ( <lettre> | <chiffre> | "-" | "_" | "*" ) [ <pnrest> ]
<nombre> = <chiffre> [ <nombre> ] -- jusqu'à une valeur décimale maximale de 4g-1
<hstr> = "0x" <hrest> -- jusqu'à une valeur décimale maximale de 4g-1
<hrest> = <hdigit> [ <hrest> ]
<lf> = caractère saut à la ligne
<cr> = caractère retour chariot
<eoln> = <cr><lf> | <lf>
<sp> = " "
<tab> = "  "
<espace> = { <sp> | <tab> | <eoln> } [ <espace> ]
<chaîne> = "" [ <strest> ] ""
<strest> = ( <lettre> | <chiffre> | <espace> ) [ <strest> ]

```

Voici la notation BNF étendue pour la grammaire avec le symbole de début <piFile> :

-- un fichier contenant une ou plusieurs définitions d'identifiant de protocole (PI, *Protocol Identifier*)

<piFile> = <piDefinition>...

-- une définition PI

```

<piDefinition> =
  <protoName> "IDENTIFIANT DE PROTOCOLE"
  [ "VARIANT-OF" <protoName> ]
  "PARAMETERS" "{" [ <parmList> ] "}"
  "ATTRIBUTES" "{" [ <attrList> ] "}"
  "DESCRIPTION" chaîne
  [ "CHILDREN" chaîne ]
  [ "ADDRESS-FORMAT" chaîne ]
  [ "DECODING" chaîne ]
  [ "REFERENCE" chaîne ]
  "::<=" "{" <encapList> "}"

```

-- un nom de protocole

<protoName> = pname



```

-- une liste de paramètres
<parmList> = <parm> [ "," <parm> ]...

-- un paramètre
<parm> = lcnme [<espace>] "(" [<espace>] <nonNegNum> [<espace>] ")" [<espace>]

-- une liste d'attributs
<attrList> = <attr> [ [<espace>] "," [<espace>] <attr> ]...

-- un attribut
<attr> = lcnme [<espace>] "(" [<espace>] <nonNegNum> [<espace>] ")"

-- un nombre non négatif
<nonNegNum> = nombre | hstr

-- une liste de valeurs d'encapsulation
<encapList> = <encapValue> [ [<espace>] "," [<espace>] <encapValue> ]...

-- une valeur d'encapsulation
<encapValue> = <baseEncapValue> | <normalEncapValue>

-- une valeur d'encapsulation de base
<baseEncapValue> = <nonNegNum>

-- une valeur d'encapsulation normale
<normalEncapValue> = <protoName> <espace> <nonNegNum>

-- un commentaire
  <deux tirets> <texte> <fin de ligne>

```

### 3.2.4 Transposition du nom de protocole

La valeur "protoName", appelée le "nom de protocole" devra être une chaîne ASCII consistant en un maximum de 64 caractères parmi les suivants : "A" à "Z", "a" à "z", "0" à "9", tiret (-), souligné (\_), astérisque (\*), plus (+)

Le premier caractère du nom de protocole est limité à un des suivants : "A" à "Z", "a" à "z", "0" à "9".

Cette valeur DEVRAIT être le nom ou l'acronyme identifiant le protocole. Noter que la casse est significative. La valeur choisie pour le nom du protocole DEVRAIT correspondre au nom ou acronyme "le plus connu" du protocole indiqué. Par exemple, le document indiqué par l'URL : <ftp://ftp.isi.edu/in-notes/iana/assignments/protocol-numbers> définit des valeurs de champ du protocole IP, de sorte que les macros d'identifiant de protocole pour des enfants de IP DEVRAIENT recevoir des noms cohérents avec les noms de protocoles qui se trouvent dans ce document d'autorité. De même, des enfants de UDP et TCP DEVRAIENT recevoir des noms cohérents avec les allocations de nom de numéro d'accès trouvés dans <ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers>.

Lorsque le "nom bien connu" contient des caractères non permis dans les noms de protocoles, ils DOIVENT être changés en un caractère tiret ("-"). Au cas où le premier caractère doit être changé, le nom du protocole est précédé de la lettre "p", de sorte que l'ancienne première lettre peut être changée en un tiret.

Par exemple, z39.50 devient z39-50 et 914c/g devient 914c-g. Les noms de protocoles suivants sont légaux :

```
ftp, ftp-data, whois++, sql*net, 3com-tsmux, ocs_cmu
```

Noter qu'il est possible dans une mise en œuvre réelle que différentes encapsulations du même protocole (qui sont représentées par différentes entrées dans le tableau protocolDirTable) se verront allouer le même nom de protocole. La valeur d'INDEX protocolDirID définit un protocole particulier, et non la chaîne du nom de protocole.

### 3.2.5 Transposition de la clause VARIANT-OF

Cette clause n'est présente que pour les protocoles alloués par l'IANA. Elle identifie la macro protocol-identifiant qui représente au plus près ce protocole particulier, et est connue comme "protocole de référence". Une macro protocol-identifiant DOIT exister pour le protocole de référence. Lorsque cette clause est présente dans une macro protocol-identifiant,

la macro est appelée un 'identifiant de variante de protocole'.

Toute clause (par exemple CHILDREN, ADDRESS-FORMAT) dans la macro protocol-identifiant de référence NE DEVRAIT PAS être dupliquée dans la macro identifiant de variante de protocole, si la sémantique des protocoles 'variants' est identique pour une certaine clause.

Comme les clauses PARAMETERS et ATTRIBUTES DOIVENT être présentes dans un identifiant de protocole, des 'ParamList' et 'AttrList' vides (c'est-à-dire, "PARAMETERS {}") DOIVENT être présentes dans une macro protocol-variant-identifiant, et la 'ParamList' ainsi que la 'AttrList' trouvées dans la macro protocol-identifiant de référence sont examinées à la place.

Noter que si un protocole 'ianaAssigned' est défini qui n'est une variante d'aucun autre protocole documenté, la macro protocol-identifiant DEVRAIT alors être utilisée à la place de la version protocol-variant-identifiant de la macro.

### 3.2.6 Transposition de la clause PARAMETERS

L'objet protocolDirParameters fournit à un NMS la capacité d'activer et désactiver de coûteuses ressources de détection. Un agent peut prendre en charge un certain paramètre tout le temps, pas du tout, ou selon la charge actuelle des ressources.

La clause PARAMETERS est une liste de définitions de bits qui peut être directement codée dans l'octet ProtocolDirParameters associé dans l'ordre des octets du réseau. Zéro, une ou plusieurs définitions de bits peuvent être présentes. Seuls les bits 0 à 7 sont des valeurs valides de codage. Cette clause définit le jeu complet de BIT permis pour un certain protocole. Un agent conforme peut choisir de mettre en œuvre un sous ensemble de zéro, un ou plusieurs des ces PARAMETERS.

Par convention, les définitions de bits communes suivantes sont utilisées par les différents protocoles. Ces positions de bits NE DOIVENT PAS être utilisées pour d'autres paramètres. Elles DOIVENT être réservées si non utilisées par un certain protocole.

Les bits sont codé dans un seul octet. Le bit 0 est le bit de poids fort (le plus à gauche) dans l'octet, et le bit 7 est le bit de moindre poids (le plus à droite) dans le premier octet. Les bits réservés et les bits non spécifiés dans l'octet sont mis à zéro.

**Tableau 3.1 : Bits PARAMETERS réservés**

Nom du bit	Description
0 countsFragments	Les protocoles de couche supérieure encapsulés dans ce protocole seront comptés correctement même si ce protocole fragmente les couches supérieures en plusieurs paquets.
1 tracksSessions	Attribue correctement tous les paquets d'un protocole qui commence les sessions sur des accès ou prises bien connus et les transfère ensuite à des accès ou prises alloués dynamiquement (par exemple TFTP).

La clause PARAMETERS DOIT être présente dans toutes les déclarations de macro protocol-identifiant, mais peut être égale à zéro (vide).

#### 3.2.6.1 Transposition du bit 'countsFragments(0)'

Ce bit indique si la sonde attribue correctement tous les paquets fragmentés du protocole spécifié, même si les trames individuelles qui portent ce protocole ne peuvent être identifiées comme telles. Noter que la sonde n'est pas obligée de présenter réellement de datagramme réassemblé (pour l'analyse d'adresse, le filtrage, ou tout autre objet) au NMS.

Ce bit DOIT n'être établi que dans un octet protocolDirParameters qui correspond à un protocole qui accepte une certaine forme de fragmentation et de réassemblage. Noter que les paquets TCP ne sont pas considérés comme des 'flux fragmentés' et donc TCP n'est pas éligible.

Ce bit PEUT être établi dans plus d'un octet protocolDirParameters au sein d'un INDEX protocolDirTable, dans le cas où un agent pourrait compter les fragments à plus d'une couche de protocole.

#### 3.2.6.2 Transposition du bit 'tracksSessions(1)'

Le bit 'tracksSessions(1)' indique si les trames qui font partie des sessions retransposées (par exemple sessions TFTP téléchargées) sont comptées correctement par la sonde. Pour un tel protocole, la sonde doit habituellement analyser tous les paquets reçus sur l'interface indiquée, et conserver des informations d'état, (par exemple le numéro d'accès UDP

retransposé pour TFTP).

La sémantique du paramètre 'tracksSessions' est indépendante des autres définitions de protocolDirParameters, de sorte que ce paramètre PEUT être combiné avec toute autre configuration légale de paramètre.

### 3.2.7 Transposition de la clause ATTRIBUTES

L'objet protocolDirType fournit à un NMS une indication des capacités d'une sonde pour décoder un certain protocole, ou les attributs généraux de ce protocole particulier.

La clause ATTRIBUTES est une liste des définitions de bits qui sont codés dans l'instance associée de ProtocolDirType. Les définitions de BIT sont spécifiées dans la clause SYNTAX de l'objet de MIB protocolDirType.

**Tableau 3.2 : Bits ATTRIBUTES réservés**

Nom du bit	Description
0 hasChildren	indique qu'il peut y avoir des enfants de ce protocole définis dans la protocolDirTable (par l'agent ou par le gestionnaire).
1 addressRecognitionCapable	indique que ce protocole peut être utilisé pour générer des entrées de tableau d'hôte et de matrice.

La clause ATTRIBUTES DOIT être présente dans toutes les déclarations de macro protocol-identifier, mais PEUT être vide.

### 3.2.8 Transposition de la clause DESCRIPTION

La clause DESCRIPTION fournit une description textuelle du protocole identifié par cette macro. Noter qu'elle NE DEVRAIT PAS contenir de détails sur les éléments couverts par les clauses CHILDREN, ADDRESS-FORMAT, DECODING et REFERENCE.

La clause DESCRIPTION DOIT être présente dans toutes les déclarations de macro protocol-identifier.

### 3.2.9 Transposition de la clause CHILDREN

La clause CHILDREN fournit une description des protocoles fils pour les protocoles qui les prennent en charge. Elle a trois sous-sections :

- Détails sur la ou les valeurs de champ utilisées pour choisir le protocole fils, et comment est effectué le processus de choix ;
- Détails sur la façon dont la ou les valeurs sont codées dans la chaîne d'octets d'identifiant de protocole ;
- Détails sur la façon dont les protocoles fils sont nommés par rapport à leurs étiquettes de protocole parent.

La clause CHILDREN DOIT être présente dans toutes les déclarations de macro de protocol-identifier dans lesquelles le bit 'hasChildren(0)' est établi dans la clause ATTRIBUTES.

### 3.2.10 Transposition de la clause ADDRESS-FORMAT

La clause ADDRESS-FORMAT donne une description du ou des formats de CHAÎNE D'OCTETS utilisés lors du codage des adresses.

Cette clause DOIT être présente dans toutes les déclarations de macro de protocol-identifier dans lesquelles le bit 'addressRecognitionCapable(1)' est établi dans la clause ATTRIBUTES.

### 3.2.11 Transposition de la clause DECODING

La clause DECODING donne une description de la procédure de décodage pour le protocole spécifié. Elle contient des conseils de décodage utiles pour la mise en œuvre, mais NE DEVRAIT PAS surdupliquer les informations dans les documents cités dans la clause REFERENCE. Elle peut contenir une description complète de toutes informations de décodage requises.

Pour les protocoles 'extensibles' (qui ont le bit 'hasChildren(0)' établi) cela inclut des informations de décalage et de type pour le ou les champs utilisés pour le choix des enfants ainsi que des informations sur la détermination du début du

protocole fils.

Pour les protocoles 'addressRecognitionCapable' (*capables de reconnaissance d'adresse*) cela inclut les informations de décalage et de type pour le ou les champs utilisés pour générer les adresses.

La clause DECODING est facultative, et PEUT être omise si la clause REFERENCE contient des pointeurs sur les informations de décodage pour le protocole spécifié.

### 3.2.12 Transposition de la clause REFERENCE

Si il existe un document de référence disponible au public pour ce protocole, il DEVRAIT être mentionné ici. Cela sera normalement un URL si possible ; sinon, ce sera le nom et l'adresse de l'organisme qui le contrôle.

Les clauses CHILDREN, ADDRESS-FORMAT, et DECODING DEVRAIENT limiter la quantité d'informations qui peuvent être obtenues d'un document d'autorité, tel que le document des numéros alloués [RFC1700]. Toute duplication ou paraphrase des informations devrait être brève et cohérente avec le document d'autorité.

La clause REFERENCE est facultative, mais DEVRAIT être mise en œuvre si il existe une référence d'autorité pour le protocole (en particulier pour les protocoles de norme).

## 3.3 Évaluation d'un indice du tableau ProtocolDir

L'évaluation suivante est faite après qu'une valeur d'INDEX de protocolDirTable a été convertie en deux CHAINES D'OCTETS conformément aux règles de codage d'INDEX spécifiées dans le SMI [RFC1902].

Les identifiants de protocole sont évalués de gauche à droite, en commençant par le protocolDirID, dont la longueur DOIT être divisible par quatre. La longueur de protocolDirParameters DOIT être exactement un quart de la longueur de la chaîne protocolDirID.

L'analyse de l'identifiant de protocole commence par l'identifiant de couche de base, qui DOIT être présent, et continue par un ou plusieurs identifiants de couche supérieure, jusqu'à ce que tous les OCTETS du protocolDirID aient été utilisés. Des couches NE DOIVENT PAS être sautées, de sorte que des identifiants comme 'SNMP sur IP' ou 'TCP sur ether2' ne doivent pas exister.

L'identifiant de couche de base contient aussi un 'identifiant de fonction spéciale' qui peut s'appliquer au reste de l'identifiant de protocole.

L'inclusion de caractères génériques (*wild-carding*) à la couche de base au sein d'une encapsulation de protocole est la seule fonction spécial acceptée pour l'instant. (Voir les détails au paragraphe 4.1.1.2.)

Après l'analyse de la chaîne Identifiant de protocole (qui est la valeur de protocolDirID) chaque octet de la chaîne Paramètres de protocole est évalué, et appliqué à la couche de protocole correspondante.

Une étiquette d'identifiant de protocole PEUT se transposer en plus d'une valeur. Par exemple, 'ip' se transpose en cinq valeurs distinctes, une pour chaque encapsulation prise en charge. (Voir la section 'IP' sous "Identifiants de protocole de couche 3" dans le document des macros d'identifiant de protocole RMON [RFC2896]).

Il est important de noter que ces macros sont conceptuellement développées au moment de la mise en œuvre, et non au démarrage.

Si toutes les macros sont complètement développées en substituant toutes les valeurs possibles de chaque étiquette pour chaque protocole fils, une liste de tous les identifiants de protocole possibles est produite. De sorte que 'ip' résulterait en cinq identifiants de protocole distincts. De même, chaque enfant de 'ip' se transposerait en au moins cinq identifiants de protocole, un pour chaque encapsulation (par exemple ip sur ether2, ip sur LLC, etc.).

## 4. Macros d'identifiant de protocole de couche de base

Les macros IDENTIFIANT DE PROTOCOLE suivantes peuvent être utilisées pour la construction protocolDirID et les chaînes protocolDirParameters.

Un identifiant est codé en construisant l'identifiant de base, puis en ajoutant un identifiant de couche pour chaque protocole encapsulé.

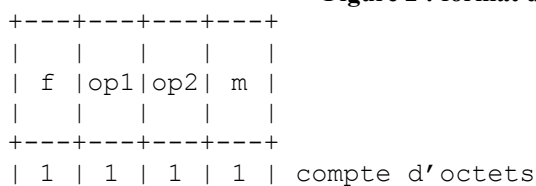
Se référer au document Macros d'identifiant de protocole RMON [RFC2896] pour la liste des macros de PI de couche non de base publiée par le groupe de travail. Noter que d'autres documents de macros de PI peuvent exister, et il devrait être possible à une mise en œuvre de remplir le tableau protocolDirTable sans utiliser le document Macro PI [RFC2896].

#### 4.1 Codage d'identifiant de base

La première encapsulation de couche est appelée l'identifiant de base et elle contient des informations facultatives de fonction de protocole et la valeur d'énumération de la couche de base (par exemple, la couche MAC) utilisée dans cet identifiant de protocole.

L'identifiant de base est codé sur quatre octets comme le montre la figure 2.

**Figure 2 : format d'identifiant de base**



Le premier octet ('f') est le code de fonction spéciale, qu'on trouve au tableau 4.1. Les deux octets suivants ('op1' et 'op2') sont les opérandes pour la fonction indiquée. Si il n'est pas utilisé, un opérande doit être réglé à zéro. Le dernier octet, 'm', est la valeur numérique d'une encapsulation de couche de base particulière, qu'on trouve au tableau 4.2. Les quatre octets sont codés dans l'ordre des octets du réseau.

##### 4.1.1 Fonctions d'identifiant de protocole

L'identifiant de couche de base contient des informations sur toutes les fonctions spéciales à effectuer durant les collectes de ce protocole, ainsi que l'identifiant d'encapsulation de couche de base.

Les trois premiers octets de l'identifiant contiennent le code de fonction et deux opérandes facultatifs. Le quatrième octet contient l'encapsulation de couche de base particulière utilisée dans ce protocole (figure 2).

**Tableau 4.1 : Fonctions d'identifiant de protocole allouées**

Fonction	ID	Param1	Param2
aucune	0	non utilisé (0)	non utilisé (0)
caractère générique	1	non utilisé (0)	non utilisé (0)

##### 4.1.1.1 Fonction 0 : aucune

Si le champ Identifiant de fonction (1<sup>er</sup> octet) est égal à zéro, les champs 'op1' et 'op2' (second et troisième octets) doivent aussi être égaux à zéro. Cette valeur spéciale indique qu'aucune fonction n'est appliquée à l'identifiant de protocole codé dans les octets restants. L'identifiant représente une encapsulation normale de protocole.

##### 4.1.1.2 Fonction 1 : fonction générique de protocole

La fonction générique (identifiant de fonction = 1) est utilisée pour agréger des compteurs, en utilisant une seule valeur de protocole pour indiquer des encapsulations potentiellement nombreuses de couches de base d'un protocole de couche réseau particulier. Une protocolDirEntry de ce type va correspondre à toute encapsulation de couche de base du même protocole de couche réseau.

Le champ 'op1' (second octet) n'est pas utilisé et DOIT être réglé à zéro.

Le champ 'op2' (troisième octet) n'est pas utilisé et DOIT être réglé à zéro.

Chaque identifiant générique de protocole DOIT être défini dans les termes d'une 'encapsulation de base'. Ceci DEVRAIT être aussi 'standard' que possible pour les besoins de l'interopérabilité. La valeur de couche de base la plus petite possible DEVRAIT être choisie. De sorte que si une encapsulation sur 'ether2' est permise, cela devrait alors être utilisé comme l'encapsulation de base. Sinon, une encapsulation sur LLC devrait alors être utilisée, si cela est permis. Et ainsi de suite pour chacune des couches de base définies.

On devrait noter qu'un agent n'a pas à prendre en charge l'identifiant de protocole non générique sur la même couche de base. Par exemple, un anneau à jetons pour seul appareil ne devrait normalement pas prendre en charge IP sur la couche de base ether2. Néanmoins, il devrait utiliser la couche de base ether2 pour définir l'encapsulation IP générique. L'agent PEUT aussi prendre en charge le comptage de certaines ou de toutes les encapsulations individuelles pour les mêmes protocoles, en plus du comptage générique. Noter que la MIB RMON-2 [RFC2021] n'exige pas que les agents entretiennent des compteurs pour plusieurs encapsulations du même protocole. C'est une question spécifique de la mise en œuvre de décider comment un agent détermine quelle combinaisons de protocoles permettre dans le protocolDirTable à chaque instant.

## 4.2 Identifiants de protocole de couche de base

La couche de base est obligatoire, et définit l'encapsulation de base du paquet et toutes les fonctions spéciales pour cet identifiant.

Aucun bit protocolDirParameters n'est suggéré pour la couche de base.

La valeur suggérée pour le champ ProtocolDirDescr pour la couche de base est donnée par le champ "Nom" correspondant dans le tableau 4.2 ci-dessous. Cependant, les mises en œuvre ne sont obligées d'utiliser que les valeurs entières appropriées d'identifiant.

Pour la plupart des protocoles de couche de base, le champ protocolDirType devrait contenir des bits établis pour les attributs 'hasChildren(0)' et 'addressRecognitionCapable(1)'. Cependant, la couche de base spéciale 'ianaAssigned' ne devrait avoir aucun bit de paramètre ou d'attribut établi.

Par conception, on ne peut prendre en charge que 255 différentes encapsulations de couche de base. Cinq valeurs d'encapsulation de base sont définies pour l'heure. On s'attend à ce que très peu de nouvelles encapsulations de base (par exemple, pour de nouveaux types de supports) soient ajoutées à l'avenir.

**Tableau 4.2 : Valeurs de codage de couche de base**

Nom	Identifiant
ether2	1
llc	2
snap	3
vsnap	4
ianaAssigned	5

### -- Encapsulation Ether2

IDENTIFIANT DE PROTOCOLE -ether2

PARAMETERS { }

ATTRIBUTS { hasChildren(0), addressRecognitionCapable(1) }

DESCRIPTION : "DIX Ethernet, aussi appelée Ethernet-II."

ENFANTS : "Le champ de type Ethernet-II est utilisé pour choisir les protocoles fils. C'est un champ de 16 bits. Les protocoles fils sont réputés commencer au premier octet après ce champ de type.

Les enfants de ce protocole sont codés comme [ 0.0.0.1 ], l'identifiant de protocole pour 'ether2' suivi par [ 0.0.a.b ] où 'a' et 'b' sont le codage dans l'ordre des octets du réseau de l'octet de poids fort et de l'octet de moindre poids de la valeur du type Ethernet-II.

Par exemple, une valeur de fragment de protocolDirID de : 0.0.0.1.0.0.8.0 définit l'encapsulation IP dans ether2.

Les enfants de ether2 sont désignés par 'ether2' suivi par la valeur du champ de type en hexadécimal. L'exemple ci-dessus deviendrait : ether2 0x0800"

FORMAT-D'ADRESSE : "Les adresses Ethernet font 6 octets dans l'ordre du réseau."

DECODAGE "Seules les valeurs de type supérieures à 1500 (décimal) indiquent des trames Ethernet-II ; les valeurs inférieures indiquent l'encapsulation 802.3 (voir ci-dessous)."

REFERENCE "La liste d'autorité des valeurs de type Ether est identifiée par l'URL :

<ftp://ftp.isi.edu/in-notes/iana/assignments/ethernet-numbers> "

::= { 1 }

## -- Encapsulation LLC

IDENTIFIANT DE PROTOCOLE llc

PARAMETERS { }

ATTRIBUTS { hasChildren(0), addressRecognitionCapable(1) }

DESCRIPTION : "Protocole de contrôle de liaison logique (LLC) 802.2."

ENFANTS : "Le point d'accès de service source (SSAP, *Source Service Access Point*) et le point d'accès de service de destination (DSAP, *Destination Service Access Point*) de LLC sont utilisés pour choisir des protocoles fils. Chacun d'eux fait un octet, bien que le bit de moindre poids soit un bit de contrôle et devrait être masqué dans la plupart des situations. Normalement, SSAP et DSAP (une fois masqués) sont les mêmes pour un protocole donné – chaque extrémité sachant implicitement si il est le serveur ou le client dans un protocole client/serveur. C'est cependant une simple convention, et il est possible que ce soit différent. Le SSAP est confronté d'abord aux protocoles fils. Si aucun n'est trouvé, c'est alors le DSAP qui leur est confronté. Le protocole fils est réputé commencer au premier octet après le ou les champs de commande de LLC.

Les enfants de 'llc' sont codés comme [ 0.0.0.2 ], le composant d'identifiant de protocole pour LLC suivi par [ 0.0.0.a ] où 'a' est la valeur de SAP qui se transpose en protocole fils. Par exemple, une valeur de protocolDirID-fragment de : 0.0.0.2.0.0.0.240 définit NetBios sur LLC.

Les enfants sont nommés 'llc' suivi par la valeur SAP en hexadécimal. L'exemple ci-dessus serait nommé : llc 0xf0"

FORMAT-D'ADRESSE : "L'adresse consiste en 6 octets d'adresse MAC dans l'ordre du réseau. Les bits d'acheminement de source devraient être supprimés de l'adresse si ils sont présents".

DÉCODAGE : "Remarquer que LLC a un en-tête de protocole de longueur variable ; il y a toujours trois octets (DSAP, SSAP, contrôle). Selon la valeur des bits de contrôle dans les champs DSAP, SSAP et contrôle, il peut y avoir un octet supplémentaire d'informations de contrôle.

LLC peut être présent sur plusieurs supports différents. Pour 802.3 et 802.5 sa présence est rendue obligatoire (mais voir les encapsulations ether2 et 802.3 brut). Pour 802.5, il n'y a pas d'autre protocole de couche liaison.

Remarquer aussi que le protocole de couche liaison raw802.3 peut prendre la préséance sur celui-ci d'une façon spécifique du protocole telle qu'il ne soit pas possible d'utiliser toutes les valeurs de LSAP si raw802.3 est aussi présent."

REFERENCE : "La liste d'autorité des valeurs LSAP de LLC est contrôlée par l'autorité d'enregistrement de l'IEEE :

IEEE Registration Authority

c/o Iris Ringel

IEEE Standards Dept

445 Hoes Lane, P.O. Box 1331

Piscataway, NJ 08855-1331

Phone +1 908 562 3813

Fax: +1 908 562 1571"

::= { 2 }

## -- Encapsulation SNAP sur LLC , (OUI (*Organizationally Unique Identifier*) =000) --

IDENTIFIANT DE PROTOCOLE snap

PARAMETERS { }

ATTRIBUTES { hasChildren(0), addressRecognitionCapable(1) }

DESCRIPTION : "Le protocole d'accès de sous-réseau (SNAP, *Sub-Network Access Protocol*) est mis en couches par dessus le protocole de LLC, permettant aux protocoles Ethernet-II de fonctionner sur un support restreint à LLC."

ENFANTS : "Les enfants de 'snap' sont identifiés par des valeurs de type Ethernet-II ; le champ SNAP Identifiant de protocole (PID) est utilisé pour choisir l'enfant approprié. L'en-tête de protocole SNAP entier est utilisé ; le protocole fils est supposé commencer au prochain octet après le PID.

Les enfants de 'snap' sont codés comme [ 0.0.0.3 ], l'identifiant de protocole pour 'snap', suivi par [ 0.0.a.b ] où 'a' et 'b' sont l'octet de poids fort et l'octet de moindre poids de la valeur de type Ethernet-II.

Par exemple, une valeur de protocolDirID-fragment de : 0.0.0.3.0.0.8.0 définit le protocole IP/SNAP.

Les enfants de ce protocole sont nommé 'snap' suivi par la valeur de type Ethernet-II en hexadécimal. L'exemple ci-dessus serait nommé : snap 0x0800"

FORMAT D'ADRESSE : "Le format d'adresse pour SNAP est le même que pour LLC"

DECODAGE : "SNAP n'est présent que sur LLC. SSAP et DSAP seront tous deux 0xAA et un seul octet de contrôle sera présent. Il y a alors trois octets d'identifiant unique d'organisation (OUI) et deux octets de PID. Pour cette encapsulation le OUI doit être 0x000000 (voir 'vsnap' ci-dessous pour les OUI non zéro)."

REFERENCE : "Les valeurs d'identifiant SNAP sont allouées par l'office des standards de l'IEEE. L'adresse est :

IEEE Registration Authority

c/o Iris Ringel

IEEE Standards Dept  
 445 Hoes Lane, P.O. Box 1331  
 Piscataway, NJ 08855-1331  
 téléphone +1 908 562 3813  
 Fax: +1 908 562 1571"  
 ::= { 3 }

-- **Encapsulation Fabricant SNAP sur LLC (OUI != 000)**

IDENTIFIANT DE PROTOCOLE vsnap  
 PARAMETERS { }  
 ATTRIBUTES { hasChildren(0), addressRecognitionCapable(1) }  
 DESCRIPTION : "Ce pseudo-protocole traite tous les paquets SNAP qui n'ont pas un OUI de zéro. Voir 'snap' ci-dessus pour les détails de ceux qui ont une valeur de OUI de zéro".  
 ENFANTS : "Les enfants de 'vsnap' sont choisis par les trois octets du OUI ; le PID n'est pas analysé ; les protocoles fils sont réputés commencer avec le premier octet du champ PID de SNAP, et continuer jusqu'à la fin du paquet. Les enfants de 'vsnap' sont codés comme [ 0.0.0.4 ], l'identifiant de protocole pour 'vsnap', suivi par [ 0.a.b.c ] où 'a', 'b' et 'c' sont les trois octets du champ OUI dans l'ordre des octets du réseau.  
 Par exemple, une valeur de protocolDirID-fragment de : 0.0.0.4.0.8.0.7 définit le jeu des protocoles spécifiques de Apple sur vsnap.  
 Les enfants sont nommés 'vsnap <OUI>', où le champ '<OUI>' est représenté par trois octets en notation hexadécimale.  
 De sorte que l'exemple ci-dessus serait nommé : 'vsnap 0x080007'"  
 FORMAT D'ADRESSE : "Le format d'adresse LLC est hérité par 'vsnap'. Voir l'identifiant de protocole 'llc' pour les détails."  
 DECODAGE : "Le même que pour 'snap' sauf que le OUI n'est pas zéro et l'identifiant de protocole SNAP n'est pas analysé".  
 REFERENCE : "Les valeurs d'identifiant SNAP sont allouées par l'office des normes de l'IEEE. L'adresse est :  
 IEEE Registration Authority  
 c/o Iris Ringel  
 IEEE Standards Dept  
 445 Hoes Lane, P.O. Box 1331  
 Piscataway, NJ 08855-1331  
 téléphone +1 908 562 3813  
 Fax: +1 908 562 1571"  
 ::= { 4 }

-- **Protocoles alloués par l'IANA**

IDENTIFIANT DE PROTOCOLE ianaAssigned  
 PARAMETERS { }  
 ATTRIBUTES { }  
 DESCRIPTION : "Cette branche contient des protocoles qui ne se conforment pas facilement au format hiérarchique utilisé dans les autres branches de couches de liaison. Habituellement, un tel protocole se conforme "presque" à un format particulier 'bien connu' d'identifiant, mais des critères supplémentaires sont utilisés (par exemple, fondés sur la configuration) qui rendent difficile ou impossible l'identification du protocole par l'examen du trafic réseau approprié (empêchant toute macro d'identifiant de protocole "bien connu" d'être utilisée).  
 Les protocoles bien connus sont parfois simplement retransposés sur un numéro d'accès différent par un ou plusieurs distributeurs (par exemple SNMP). Ces protocoles peuvent être identifiés avec la caractéristique d'extensibilité limitée' du protocolDirTable, et n'exigent pas d'allocations spéciales de l'IANA.  
 Une liste localisée centralement de ces protocoles énumérés doit être tenue par l'IANA pour assurer l'interopérabilité. (Voir au paragraphe 2.3 les détails de la procédure de mise à jour du document.) La prise en charge de nouvelles couches de liaison sera ajoutée explicitement, et seuls les protocoles qui ne peuvent pas être représentés d'une meilleure façon seront considérés comme des protocoles 'ianaAssigned'.  
 Les protocoles IANA sont identifiés par la valeur de sélecteur de couche de base [ 0.0.0.5 ], suivie par les quatre octets [ 0.0.a.b ] de la valeur d'entier correspondant au protocole IANA en question.  
 Ne pas créer d'enfant de ce protocole si on n'est pas sûr qu'ils ne peuvent pas être traités par les couches de liaison conventionnelles ci-dessus."  
 ENFANTS : "Les enfants de ce protocole sont identifiés par des moyens spécifiques de la mise en œuvre, décrits (aussi bien que possible) dans la clause 'DECODAGE' au sein de la macro identifiant de variante de protocole pour chaque protocole énuméré.  
 Les enfants de ce protocole sont codés comme [ 0.0.0.5 ], l'identifiant de protocole pour 'ianaAssigned', suivi par [ 0.0.a.b ] où 'a', 'b' sont les codages dans l'ordre du réseau de l'octet de poids fort et de l'octet de moindre poids de la valeur citée pour le protocole particulier alloué par l'IANA.



Par exemple, une valeur de protocolDirID-fragment de : 0.0.0.5.0.0.0.1 définit le protocole IPX encapsulé directement dans 802.3.

Les enfants sont nommés 'ianaAssigned' suivi par la valeur numérique du protocole particulier alloué par l'IANA. L'exemple ci-dessus serait nommé : 'ianaAssigned 1' "

DECODAGE : "La couche de base 'ianaAssigned' est un pseudo-protocole et n'est pas décodé."

REFERENCE : "Se référer aux macros individuelles d'IDENTIFIANT DE PROTOCOLE pour les informations sur chaque enfant du protocole alloué par l'IANA."

```
::= { 5 }
```

-- Les déclarations de macro d'identifiant de variante de protocole suivantes sont utilisées pour identifier les protocoles RMONMIB alloués par l'IANA d'une façon propriétaire, par simple énumération.

IDENTIFIANT DE PROTOCOLE ipxOverRaw8023

VARIANT-OF ipx

PARAMETERS { }

ATTRIBUTES { }

DESCRIPTION : "Ce pseudo-protocole décrit une encapsulation de IPX sur 802.3, sans champ Type. Se référer à la macro pour IPX pour des informations supplémentaires sur ce protocole."

DECODAGE : "Chaque fois que l'en-tête 802.3 indique LLC, un ensemble d'essais spécifiques du protocole doit être appliqué pour déterminer si c'est un paquet 'raw8023' ou un vrai paquet 802.2. La nature de ces essais dépend des protocoles fils actifs pour 'raw8023' et sort du domaine d'application du présent document."

```
::= {
```

```
ianaAssigned 1,          -- [0.0.0.1]
```

```
802-1Q    0x05000001    -- 1Q_IANA [5.0.0.1]
```

```
}
```

### 4.3 Couches d'encapsulation

Les couches d'encapsulation sont positionnées entre la couche de base et la couche réseau. C'est une question spécifique de la mise en œuvre de savoir si une sonde expose toutes ces encapsulations dans son répertoire de protocoles RMON-2.

#### 4.3.1 IEEE 802.1Q

Les sondes RMON peuvent rencontrer des trames "étiquetées VLAN" sur des liaisons de surveillance. Les normes d'encapsulation de LAN IEEE virtuel (VLAN) [IEEE802.1Q] et [IEEE802.1D-1998], définissent une couche d'encapsulation insérée après la couche MAC et avant la couche réseau. Ce paragraphe définit une macro PI qui prend en charge la plupart (mais pas la totalité) des caractéristiques de cette couche d'encapsulation.

En particulier, la macro PI RMON '802-1Q' n'expose pas le bit Encapsulation d'anneau à jeton (TR-encaps) dans la portion TCI de l'en-tête de VLAN. C'est une question spécifique de la mise en œuvre de décider si une sonde RMON convertit les trames formatées en anneau à jeton LLC (LLC-TR) en format LLC natif (LLC-N) pour les besoins de la collecte RMON.

Pour prendre en charge les formats Ethernet et LLC-N de la façon la plus efficace, et conserver quand même l'alignement avec l'approche de la couche de base RMON-2 'écroulée' (c'est-à-dire la prise en charge de snap et de vsnap) les enfants de 802.1Q sont codés un peu différemment des enfants des autres identifiants de couches de base.

IDENTIFIANT DE PROTOCOLE 802-1Q

PARAMETERS { }

ATTRIBUTES { hasChildren(0) }

DESCRIPTION : "En-tête d'encapsulation de VLAN IEEE 802.1Q. Noter que le codage spécifique du champ TPID n'est pas explicitement identifié par cette macro PI. Le codage des champs TPID codés en Ethernet ou en SNAP peut être identifié par le ifType de la source de données pour une collection RMON particulière, car le format codé en SNAP est utilisé exclusivement sur les supports d'anneaux à jetons et FDDI. Aussi, aucune information détenue dans le champ TCI (y compris le bit TR-encap) n'est identifiée dans les chaînes protocolDirID qui utilisent cette macro PI."

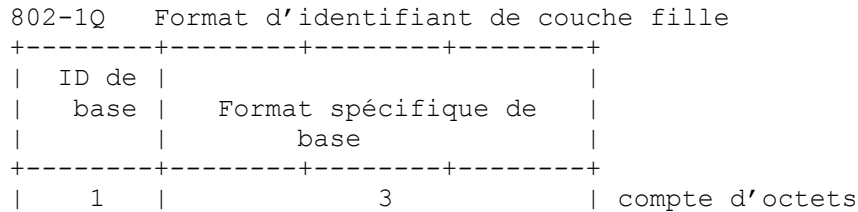
ENFANTS : "Le premier octet de l'identifiant d'enfant de quatre octets est utilisé pour distinguer le codage de base particulier qui suit l'en-tête 802.1Q. Les trois octets restants sont utilisés exactement comme défini par le codage de couche de base indiqué.

Afin de simplifier le codage d'enfants pour les cas les plus courants, les couches de base 'ether2' et 'snap' sont combinées en un seul identifiant, avec une valeur de zéro. Les autres couches de base sont codées avec des valeurs tirées du Tableau 4.2.

Valeurs d'identifiant de base 802-1Q

Couche de base	Codage du tableau 4.2	Codage de l'identifiant de base
ether2	1	0
llc	2	2
snap	3	0
vsnap	4	4ianaAssigned 5 5

Le format générique d'identifiant de couche de base est donné ci-dessous :



#### ID de base == 0

Pour les charges utiles codées avec les en-têtes soit Ethernet, soit LLC/SNAP à la suite de l'en-tête VLAN, les enfants de ce protocole sont identifiés exactement comme décrit pour les couches de base 'ether2' ou 'snap'.

Les enfants sont codés comme [ 0.0.129.0 ], l'identifiant de protocole pour '802-1Q' suivi par [ 0.0.a.b ] où 'a' et 'b' sont le codage dans l'ordre des octets du réseau de l'octet de poids fort et de l'octet de moindre poids de la valeur de type Ethernet-II.

Par exemple, une valeur de fragment de protocolDirID de 0.0.0.1.0.0.129.0.0.0.8.0 définit IP, encapsulé dans un VLAN dans ether2.

Les enfants de ce format sont désignés par '802-1Q' suivi par la valeur de champ de type en hexadécimal.

Ainsi, l'exemple ci-dessus serait déclaré comme '802-1Q 0x0800'.

#### ID de base == 2

Pour les charges utiles codées avec un en-tête de LLC (non SNAP) qui suit l'en-tête de VLAN, les enfants de ce protocole sont identifiés exactement comme décrit pour la couche de base 'llc'.

Les enfants sont codés comme [ 0.0.129.0 ], le composant d'identifiant de protocole pour 802.1Q, suivi par [ 2.0.0.a ] où 'a' est la valeur de SAP qui se transpose en le protocole fils. Par exemple, une valeur de fragment de protocolDirID de 0.0.0.1.0.0.129.0.2.0.0.240 définit NetBios, encapsulé dans un VLAN sur LLC.

Les enfants sont désignés par '802-1Q' suivi par la valeur de SAP en hexadécimal, avec l'octet de tête réglé à 2.

De sorte que l'exemple ci-dessus aurait été nommé : '802-1Q 0x020000f0'.

#### ID de base == 4

Pour les charges utiles codées avec des en-têtes LLC/SNAP (OUI non à zéro) qui suivent l'en-tête de VLAN, les enfants de ce protocole sont identifiés exactement comme décrit pour la couche de base 'vsnap'.

Les enfants sont codés comme [ 0.0.129.0 ], l'identifiant de protocole pour '802-1Q', suivi par [ 4.a.b.c ] où 'a', 'b' et 'c' sont les trois octets du champ OUI dans l'ordre des octets du réseau.

Par exemple, une valeur de fragment de protocolDirID de : 0.0.0.1.0.0.129.0.4.8.0.7 définit le jeu spécifique de protocoles Apple encapsulés dans un VLAN sur vsnap.

Les enfants sont désignés par '802-1Q' suivi par la valeur de <OUI> qui est représentée par trois octets en notation hexadécimale, avec un octet de tête réglé à la valeur de 4.

De sorte que l'exemple ci-dessus serait nommé : '802-1Q 0x04080007'.

#### ID de base == 5

Pour les charges utiles qui ne peuvent être identifiées que comme protocole 'ianaAssigned', les enfants de ce protocole sont identifiés exactement comme décrit pour la couche de base 'ianaAssigned'.

Les enfants sont codés comme [ 0.0.129.0 ], l'identifiant de protocole pour '802-1Q', suivi par [ 5.0.a.b ] où 'a' et 'b' sont le codage dans l'ordre des octets du réseau de l'octet de poids fort et de l'octet de moindre poids de la valeur d'énumération du protocole particulier allouée par l'IANA.

Par exemple, une valeur de fragment de protocolDirID de : 0.0.0.1.0.0.129.0.5.0.0.0.1 définit le protocole IPX, encapsulé

en VLAN directement en 802.3

Les enfants sont désignés par '802-1Q' suivi par la valeur numérique du protocole particulier allouée par l'IANA, avec l'octet de tête réglé à la valeur 5.

Les enfants sont désignés par '802-1Q' suivi par le codage en hexadécimal de l'identifiant de l'enfant. L'exemple ci-dessus serait nommé : '802-1Q 0x05000001'.

DECODAGE : "Les en-têtes de VLAN et la structure de trame étiquetée sont définis dans [IEEE802.1Q]."

REFERENCE : "Le protocole 802.1Q est défini dans le projet de norme sur les réseaux virtuels de zone locale pontés [IEEE802.1Q]."

```
 ::= {
     ether2 0x8100          -- Codage Ethernet ou SNAP de TPID exclu pour réduire la taille et complexité de PD.
     -- snap 0x8100
 }
```

## 5. Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## 6. Remerciements

Le présent document a été produit par le groupe de travail RMONMIB de l'IETF.

Les auteurs tiennent à remercier les personnes suivantes de leur contribution au présent document : Anil Singhal (Frontier Software Development, Inc) Jeanne Haney (Bay Networks) Dan Hansen (Network General Corp.)

Des remerciements particuliers sont adressés à David Perkins (DeskTalk Systems, Inc.) qui a écrit les compilateurs de macro RMON PI, et à Skip Koppenhaver (Technically Elite, Inc.) qui a amélioré la spécification du langage de macro PI.

## 7. Références

- [AF-LANE] LAN Emulation Sub-working Group, B. Ellington, "LAN Emulation over ATM - Version 1.0", AF-LANE-0021.000, ATM Forum, IBM, janvier 1995.
- [AF-NM-TEST] Network Management Sub-working Group, Test Sub-working Group, A. Bierman, "Remote Monitoring MIB Extensions for ATM Networks", AF-NM-TEST-0080.000, ATM Forum, Cisco Systems, février 1997.
- [IEEE802.1D] LAN MAN Standards Committee of the IEEE Computer Society, "Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks -- Common specification -- Part 3: Media Access Control (MAC) Bridges", ISO/IEC Final DIS 15802-3 (IEEE P802.1D/D17) Institute of Electrical and Electronics Engineers, Inc., mai 1998.
- [IEEE802.1Q] LAN MAN Standards Committee of the IEEE Computer Society, "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", Draft Standard P802.1Q/D11,

Institute of Electrical and Electronics Engineers, Inc., juillet 1998.

- [RFC1155] M. Rose et K. McCloghrie, "Structure et [identification des informations de gestion](#) pour les internets fondés sur TCP/IP", STD 16, mai 1990.
- [RFC1157] J. Case, M. Fedor, M. Schoffstall et J. Davin, "Protocole [simple de gestion de réseau](#)", STD 15, mai 1990. (*Historique*)
- [RFC1212] M. Rose et K. McCloghrie, "[Définitions concises de MIB](#)", STD 16, février 1991.
- [RFC1215] M. Rose, "Convention pour la définition de filtres à utiliser avec le SNMP", mars 1991. (*Info*)
- [RFC1483] Juha Heinanen, "Encapsulation multiprotocole sur couche 5 d'adaptation ATM", juillet 1993. (*Obs. voir RFC 2684*)
- [RFC1700] J. Reynolds et J. Postel, "[Numéros alloués](#)", STD 2, octobre 1994. (*Historique, voir [www.iana.org](http://www.iana.org)*)
- [RFC1901] J. Case, K. McCloghrie, M. Rose, S. Waldbusser "Introduction à SNMPv2 fondé sur la communauté", janvier 1996. (*Hist.*)
- [RFC1902] J. Case, K. McCloghrie, M. Rose, S. Waldbusser "Structure des informations de gestion pour la version 2 du protocole simple de gestion de réseau (SNMPv2)", janvier 1996. (*Obsolète, voir RFC2578*) (*D.S.*)
- [RFC1903] J. Case, K. McCloghrie, M. Rose, S. Waldbusser "Conventions textuelles pour la version 2 du protocole simple de gestion de réseau (SNMPv2)", janvier 1996. (*Obsolète, voir RFC2579*) (*D.S.*)
- [RFC1904] J. Case, K. McCloghrie, M. Rose, S. Waldbusser "Déclarations de conformité pour la version 2 du protocole simple de gestion de réseau (SNMPv2)", janvier 1996. (*Obsolète, voir RFC2580*) (*D.S.*)
- [RFC1905] J. Case, K. McCloghrie, M. Rose, S. Waldbusser "Opérations de protocole pour la version 2 du protocole simple de gestion de réseau (SNMPv2)", janvier 1996. (*Obsolète, voir RFC3416*) (*D.S.*)
- [RFC1906] J. Case, K. McCloghrie, M. Rose, S. Waldbusser "Transpositions de transport pour la version 2 du protocole simple de gestion de réseau (SNMPv2)", janvier 1996. (*Obsolète, voir RFC3417*) (*D.S.*)
- [RFC2021] S. Waldbusser, "Base de données d'informations de gestion de surveillance de réseau à distance version 2 avec SMIV2", janvier 1997. (*Obsolète, voir RFC4502*) (*P.S.*)
- [RFC2074] A. Bierman, R. Iddon, "Identifiants de protocole de MIB de surveillance de réseau à distance", janvier 1997. (*Obsolète, voir RFC2895*) (*P.S.*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2223] J. Postel, J. Reynolds, "Instructions pour les auteurs de RFC", octobre 1997. (*Information*)
- [RFC2271] D. Harrington et autres, "Architecture de description du cadre de gestion SNMP", janvier 1998. (*Obs., voir RFC2571*) (*PS*)
- [RFC2272] J. Case, D. Harrington, R. Presuhn, B. Wijnen, "Traitement et distribution de message pour le protocole simple de gestion de réseau (SNMP)", janvier 1998. (*Obsolète, voir RFC2572*) (*P.S.*)
- [RFC2273] D. Levi, P. Meyer, B. Stewart, "Applications SNMPv3", janvier 1998. (*Obsolète, voir RFC2573*) (*P.S.*)
- [RFC2274] U. Blumenthal, B. Wijnen, "Modèle de sécurité fondé sur l'utilisateur (USM) pour la version 3 du protocole simple de gestion de réseau (SNMPv3)", janvier 1998. (*Obsolète, voir RFC2574*) (*P.S.*)
- [RFC2275] B. Wijnen, R. Presuhn, K. McCloghrie, "Modèle de contrôle d'accès fondé sur la vue (VACM) pour le protocole simple de gestion de réseau (SNMP)", janvier 1998. (*Obsolète, voir RFC2575*) (*P.S.*)
- [RFC2570] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction à la version 3 du cadre de gestion de réseau de l'Internet", avril 1999. (*Obsolète, voir RFC3410*) (*Information*)
- [RFC2571] B. Wijnen, D. Harrington, R. Presuhn, "Architecture pour la description des cadres de gestion SNMP", avril 1999. (*Obsolète, voir RFC3411*) (*D.S.*)

- [RFC2572] J. Case, D. Harrington, R. Presuhn, B. Wijnen, "Traitement et répartition de message pour le protocole simple de gestion de réseau (SNMP)", avril 1999. (*Obsolète, voir [RFC3412](#)*) (D.S.)
- [RFC2573] D. Levi, P. Meyer, B. Stewart, "Applications SNMP", avril 1999. (*Obsolète, voir [RFC3413](#)*) (D.S.)
- [RFC2574] U. Blumenthal, B. Wijnen, "Modèle de sécurité fondé sur l'utilisateur (USM) pour la version 3 du protocole simple de gestion de réseau (SNMPv3)", avril 1999. (*Obsolète, voir [RFC3414](#)*) (D.S.)
- [RFC2575] B. Wijnen, R. Presuhn, K. McCloghrie, "Modèle de contrôle d'accès fondé sur la vue (VACM) pour le protocole simple de gestion de réseau (SNMP)", avril 1999. (*Obsolète, voir [RFC3415](#)*) (D.S.)
- [RFC2578] K. McCloghrie, D. Perkins, J. Schoenwaelder, "[Structure des informations de gestion](#), version 2 (SMIv2)", avril 1999. ([STD0058](#))
- [RFC2579] K. McCloghrie, D. Perkins, J. Schoenwaelder, "[Conventions textuelles pour SMIv2](#)", avril 1999. ([STD0058](#))
- [RFC2580] K. McCloghrie, D. Perkins, J. Schoenwaelder, "[Déclarations de conformité pour SMIv2](#)", avril 1999. ([STD0058](#))
- [RFC2896] A. Bierman, C. Bucci, R. Iddon, "Macros d'identifiant de protocole de la MIB de surveillance de réseau à distance", août 2000. (*Information*)

## 8. Considérations relatives à l'IANA

Les protocoles identifiés dans la présente spécification sont presque entièrement définis dans des documents externes. Dans quelques rares cas, une allocation arbitraire d'identifiant de protocole doit être faite pour la prise en charge d'un certain protocole dans le tableau protocolDirTable de RMON-2. Les macros d'identifiant de protocole pour de tels protocoles seront définies sous la couche de base 'ianaAssigned' (voir la section 3. et le paragraphe 4.2).

Pour l'instant, un seul protocole est défini sous la couche de base ianaAssigned, appelé 'ipxOverRaw8023' (voir au paragraphe 4.2).

## 9. Considérations sur la sécurité

Le présent document discute de la syntaxe et de la sémantique des descriptions textuelles des protocoles de réseautage, et non de la définition d'aucun comportement de réseautage. À ce titre, aucune considération de sécurité n'est soulevée dans le présent mémoire.

## 10. Adresse des auteurs

Andy Bierman  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA USA 95134  
téléphone : +1 408-527-3711  
mél : [abierman@cisco.com](mailto:abierman@cisco.com)

Chris Bucci  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA USA 95134  
téléphone : +1 408-527-5337  
mél : [cbucci@cisco.com](mailto:cbucci@cisco.com)

Robin Iddon  
c/o 3Com Inc.  
Blackfriars House  
40/50 Blackfriars Street  
Edinburgh, EH1 1NE, UK  
téléphone : +44 131.558.3888

## Appendice A : Changements depuis la RFC 2074

Les différences entre la RFC 2074 et le présent document sont :

- La RFC 2074 a été partagée entre un document de référence (le présent document) en cours de normalisation et un document pour information, la [RFC2896], afin de retirer la plupart des macros d'identifiant de protocole du document en cours de normalisation.
- Mises à jour administratives ; ajout d'un auteur, ajout des droits de reproduction, mise à jour du cadre SNMP passe-

- partout ;
- Mise à jour de la section Généralités.
- Le texte DOIT, DEVRAIT du paragraphe 2.1 ajouté selon le modèle ;
- De nouveaux termes sont ajoutés au paragraphe 2.1 : protocole parent, protocole fils, arborescence d'encapsulation de protocole ;
- Ajout du paragraphe 2.3 sur le partage en deux documents : "Relations avec le document des macros d'identifiant de protocole RMON".
- Ajout du paragraphe 2.4 "Relations avec la MIB ATM-RMON".
- Réécriture du paragraphe 3.2 "Format de macro d'identifiant de protocole", mais sans changement sémantique ; la syntaxe de macro PI est maintenant spécifiée plus en détails en utilisant la notation BNF.
- Au paragraphe 3.2.3.1 "Transposition du BIT 'countsFragments(0)", ce paragraphe a été précisé pour permettre plusieurs octets protocolDirParameters dans une certaine chaîne PI pour régler le bit 'countsFragments'. La version de la RFC2074 disait que juste un octet peut régler ce BIT. C'est un dispositif utile pour identifier la fragmentation à plusieurs couches, et la plupart des agents RMON-2 le faisaient déjà, de sorte que le groupe de travail a accepté cette précision.
- Ajout du paragraphe 4.3 "Couches d'encapsulation".
- Le présent document se termine après les définitions de l'encapsulation de couche de base (par le paragraphe 5.2 de la RFC2074).
- Ajout de la section sur les droits de propriété intellectuelle.
- Déplacement du paragraphe 5.3 de la RFC2074 "Couche 3 : Enfants des identifiants de protocole de base" à la fin de la RFC 2074, dans le document de référence PI [RFC2896], dans lequel ont été ajoutées de nombreuses nouvelles macros d'identifiant de protocole pour les protocoles d'application et les piles de protocole non IP.
- La section des remerciements a été mise à jour.

## 11. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2000). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de droits de reproduction ci-dessus et le présent paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou ses successeurs ou ayant droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

### Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.