

Groupe de travail Réseau
Request for Comments : 4001
 RFC rendue obsolète : 3291
 Catégorie : En cours de normalisation
 février 2005

M. Daniele, SyAM Software, Inc.
 B. Haberman, Johns Hopkins University
 S. Routhier, Wind River Systems, Inc.
 J. Schoenwaelder, International University Bremen
 Traduction Claude Brière de L'Isle

Conventions textuelles pour les adresses réseau Internet

Statut de ce mémo

Le présent document spécifie un protocole de normalisation Internet pour la communauté de l'Internet, qui appelle à la discussion et à des suggestions pour son amélioration. Prière de se reporter à l'édition en cours des "Normes de protocole officielles de l'Internet" (STD 1) sur l'état de la normalisation et le statut de ce protocole. La distribution du présent mémo n'est soumise à aucune restriction.

Notice de Copyright

Copyright (C) The Internet Society (2005).

Résumé

Le présent module de MIB définit des conventions textuelles pour représenter les informations d'adressage de couche réseau Internet couramment utilisées. L'objectif est que ces conventions textuelles soient importées et utilisées dans les modules de MIB qui devraient autrement définir leurs propres représentations.

Table des matières

Conventions textuelles pour les adresses réseau Internet.....	1
1 Introduction.....	1
2 Le cadre de gestion de la norme Internet.....	3
3 Définitions.....	3
4 Conseils d'utilisation.....	8
4.1 Indexage de tableau.....	9
4.2 Unicité des adresses.....	9
4.3 Plusieurs adresses par hôte.....	9
4.4 Résolution des noms DNS.....	9
5 Exemple d'indexage de tableau.....	10
6 Considérations sur la sécurité.....	11
7 Remerciements.....	11
8 Changements de la RFC 3291 à la RFC 4001.....	11
9 Changements de la RFC 2851 à la RFC 3291.....	12
10 Références.....	12
10.1 Références normatives.....	12
10.2 Références informatives.....	12

1 Introduction

Plusieurs modules de MIB en cours de normalisation utilisent le type de base de SMIV2 IpAddress. Ceci limite l'applicabilité de ces modules de MIB à IP Version 4 (IPv4), car le type de base de SMIV2 IpAddress ne peut contenir que des adresses IPv4 de 4 octets. Le type de base de SMIV2 IpAddress est devenu problématique avec l'introduction des adresses IP Version 6 (IPv6) [RFC3513].

Le présent document définit plusieurs conventions textuelles (TC) comme moyens d'exprimer les adresses génériques de couche réseau Internet au sein des spécifications de module MIB. La solution est compatible avec SMIV2 (STD 58) et SMIV1 (STD 16). Les nouvelles définitions de MIB qui doivent exprimer des adresses Internet de couche réseau DEVRAIENT utiliser les conventions textuelles définies dans le présent mémo. Les nouveaux modules MIB NE DEVRAIENT PLUS utiliser le type de base SMIV2 IpAddress.

Une adresse Internet générique consiste en deux objets : un dont la syntaxe est `InetAddressType`, et un autre dont la syntaxe est `InetAddress`. La valeur du premier objet détermine comment est codée la valeur du second. La convention textuelle `InetAddress` représente une valeur d'adresse Internet opaque. L'énumération `InetAddressType` est utilisée pour "distribuer" la valeur `InetAddress` dans une convention textuelle concrète pour le type d'adresse. Cet usage de plusieurs conventions textuelles permet l'expression des caractéristiques d'affichage de chaque type d'adresse et rend extensible l'ensemble des types d'adresses Internet défini.

Les conventions textuelles pour les domaines de transport bien connus prennent en charge les adresses Internet ciblées. La cible d'une adresse Internet est un domaine topologique au sein duquel l'adresse peut être utilisée comme identifiant univoque pour une interface ou ensemble d'interfaces. Une zone de cible (ou, simplement, une zone) est une région connectée concrète de la topologie d'une cible donnée. Noter qu'une zone est une instance particulière d'une région topologique, tandis qu'une cible est la taille d'une région topologique [RFC4007]. Comme les adresses Internet sur les appareils qui connectent plusieurs zones ne sont pas nécessairement uniques, un index de zone supplémentaire est nécessaire sur ces appareils pour choisir une interface. Les conventions textuelles `InetAddressIPv4z` et `InetAddressIPv6z` sont fournies pour prendre en charge les adresses Internet qui comportent un index de zone. Pour prendre en charge des combinaisons arbitraires d'adresses Internet ciblées, les auteurs de MIB DEVRAIENT utiliser un objet `InetAddressType` distinct pour chaque objet `InetAddress`.

Les conventions textuelles définies dans le présent document peuvent aussi être utilisées pour représenter des sous-réseaux Internet et des gammes d'adresse Internet génériques. Un sous-réseau Internet générique est représenté par trois objets : un dont la syntaxe est `InetAddressType`, un second dont la syntaxe est `InetAddress`, et un troisième dont la syntaxe est `InetAddressPrefixLength`. La valeur `InetAddressType` détermine le format concret de la valeur `InetAddress`, tandis que `InetAddressPrefixLength` identifie le préfixe d'adresse réseau Internet.

Une gamme générique d'adresses Internet consécutives est représentée par trois objets. Le premier a la syntaxe `InetAddressType`, et les objets restants ont la syntaxe `InetAddress` et spécifient le début et la fin de la gamme d'adresses. Ici aussi, la valeur `InetAddressType` détermine le format des valeurs de `InetAddress`.

Les conventions textuelles définies dans le présent document peuvent être utilisées pour définir les adresses Internet en utilisant les noms de domaine DNS en plus des adresses IPv4 et IPv6. Un concepteur de MIB peut écrire des déclarations de conformité pour exprimer que seul un sous-ensemble des types d'adresses possibles doit être pris en charge par une mise en œuvre conforme.

Les développeurs de MIB qui ont besoin de représenter les adresses Internet DEVRAIENT utiliser ces définitions chaque fois qu'elles sont applicables, et non pas définir leurs propres constructions. Même les modules MIB qui ont seulement besoin de représenter des adresses IPv4 ou IPv6 DEVRAIENT utiliser les conventions textuelles `InetAddressType/InetAddress` définies dans le présent mémo.

De nombreux modules MIB largement répandus utilisent les adresses IPv4 et ont été révisés pour prendre en charge IPv6. Ces modules MIB peuvent être rangés dans les catégories suivantes :

1. Modules MIB qui définissent des informations de gestion qui sont, en principe, insensibles à la version IP, mais le MIB utilise actuellement une construction d'adressage spécifique d'une certaine version IP.
2. Modules MIB qui définissent des informations de gestion qui sont spécifiques d'une version IP particulière (IPv4 ou IPv6) et qui ne seront vraisemblablement jamais applicables à une autre version IP.

Les modules MIB du premier type DEVRAIENT fournir des définitions d'objet (par exemple, des tableaux) qui fonctionnent avec toutes les versions d'IP. En particulier, lors de la révision d'un module MIB qui contient des tableaux spécifiques de IPv4, il est suggéré de définir de nouveaux tableaux utilisant les conventions textuelles définies dans le présent mémo qui prennent en charge toutes les versions d'IP. Le statut des nouveaux tableaux DEVRAIT être "actuel", tandis que le statut des tableaux spécifiques de la vieille version IP DEVRAIT être changé en "déconseillé". L'autre approche, consistant à avoir plusieurs tableaux similaires pour les différentes versions IP, est fortement déconseillée.

Les modules MIB du second type, qui sont par nature spécifiques d'une version IP, n'ont pas besoin d'être redéfinis. Noter que même dans ce cas, toute addition à ces modules MIB ou à des modules MIB spécifiques d'une version IP DEVRAIT utiliser les conventions textuelles définies dans le présent mémo.

Les développeurs de MIB NE DEVRAIENT PAS utiliser les conventions textuelles définies dans le présent document pour représenter des adresses génériques de couche transport. Un ensemble spécial de conventions textuelles est défini à cette fin dans la RFC 3419 [RFC3419].

Les mots clés "DOIT", "NE DOIT PAS", "DEVRAIT", "NE DEVRAIT PAS", et "PEUT", dans le présent document sont à interpréter comme décrit dans la RFC 2119 [RFC2119].

2 Le cadre de gestion de la norme Internet

Pour une vision détaillée des documents qui décrivent le cadre actuel de gestion de la norme Internet, prière de se référer à la section 7 de la [RFC3410].

On accède aux objets gérés via un magasin d'informations virtuel, appelé une base d'informations de gestion (MIB, *Management Information Base*). On accède généralement aux objets du MIB par le protocole simple de gestion de réseau (SNMP, *Simple Network Management Protocol*). Les objets dans le MIB sont définis au moyen des mécanismes définis dans la structure des informations de gestion (SMI, *Structure of Management Information*). Le présent mémo spécifie un module MIB qui est conforme à SMIV2, qui est décrite dans le STD 58 (RFC 2578, RFC 2579 et RFC 2580).

3 Définitions

INET-ADDRESS-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, mib-2, Unsigned32 FROM SNMPv2-SMI
TEXTUAL-CONVENTION FROM SNMPv2-TC;

inetAddressMIB MODULE-IDENTITY

LAST-UPDATED "200502040000Z"

ORGANIZATION

"IETF Operations and Management Area"

CONTACT-INFO

"Juergen Schoenwaelder (Editor)

International University Bremen

P.O. Box 750 561

28725 Bremen, Germany

Téléphone : +49 421 200-3587

mél : j.schoenwaelder@iu-bremen.de

Envoyer les commentaires à <ietfmbis@ops.ietf.org>."

DESCRIPTION

"Ce module MIB définit les conventions textuelles pour représenter les adresses Internet. Une adresse Internet peut être une adresse IPv4, une adresse IPv6, ou un nom de domaine DNS. Ce module définit aussi les conventions textuelles pour les numéros d'accès Internet, les numéros de systèmes autonomes, et la longueur d'un préfixe d'adresse Internet.

Copyright (C) The Internet Society (2005). Cette version de ce module MIB fait partie de la RFC 4001, voir dans la RFC elle-même les notices légales complètes."

REVISION "200502040000Z"

DESCRIPTION

"Troisième version, publiée comme RFC 4001. Cette révision introduit les conventions textuelles

InetZoneIndex, InetScopeType, and InetVersion."

REVISION "200205090000Z"

DESCRIPTION

"Seconde version, publiée comme RFC 3291. Cette révision contient plusieurs éclaircissements et introduit plusieurs nouvelles conventions textuelles :

InetAddressPrefixLength, InetPortNumber,

InetAutonomousSystemNumber, InetAddressIPv4z, et

InetAddressIPv6z."

REVISION "200006080000Z"

DESCRIPTION

"Version initiale, publiée comme RFC 2851."

::= { mib-2 76 }

InetAddressType ::= TEXTUAL-CONVENTION

STATUT actuel

DESCRIPTION

"Une valeur qui représente un type d'adresse Internet.

- unknown(0) Type d'adresse inconnu. Cette valeur DOIT être utilisée si la valeur de l'objet InetAddress correspondant est une chaîne de longueur zéro. Elle peut aussi être utilisée pour indiquer une adresse IP qui n'est pas dans un des formats définis ci-dessous.
- ipv4(1) Une adresse IPv4 définie par la convention textuelle InetAddressIPv4.
- ipv6(2) Une adresse IPv6 définie par la convention textuelle InetAddressIPv6.
- ipv4z(3) Une adresse IPv4 non mondiale comportant un index de zone comme défini par la convention textuelle InetAddressIPv4z.
- ipv6z(4) Une adresse IPv6 non mondiale comportant un index de zone comme défini par la convention textuelle InetAddressIPv6z.
- dns(16) Un nom de domaine DNS comme défini par la convention textuelle InetAddressDNS.

Chaque définition d'une valeur InetAddressType concrète doit être accompagnée d'une définition d'une convention textuelle à utiliser avec ce InetAddressType.

Pour la prise en charge des extensions futures, la convention textuelle InetAddressType NE DEVRAIT PAS être subdivisée en sous-types dans les définitions de type d'objet. Elle PEUT être subdivisée en sous-types dans des déclarations de conformité afin d'exiger seulement un sous-ensemble de ces types d'adresse pour une mise en œuvre conforme.

Les mises en œuvre doivent s'assurer que les objets InetAddressType et tout objet dépendant (par exemple, les objets InetAddress) sont cohérents. Une erreur inconsistentValue (*valeur incohérente*) doit être générée si une tentative de changement d'un objet InetAddressType devait, par exemple, conduire à une valeur d'InetAddress indéfinie. En particulier, les paires InetAddressType/InetAddress doivent être changées ensemble si le type d'adresse change (par exemple, de ipv6(2) en ipv4(1))."

```
SYNTAX      INTEGER {
unknown(0),
ipv4(1),
ipv6(2),
ipv4z(3),
ipv6z(4),
dns(16)
}
```

InetAddress ::= TEXTUAL-CONVENTION

STATUT actuel

DESCRIPTION

"Note une adresse Internet générique.

Une valeur InetAddress est toujours interprétée dans le contexte d'une valeur InetAddressType. Chaque usage de la convention textuelle InetAddress est obligé de spécifier l'objet InetAddressType qui fournit le contexte. Il est suggéré que l'objet InetAddressType soit enregistré logiquement avant le ou les objets qui utilisent la convention textuelle InetAddress, si ils apparaissent dans la même rangée logique.

La valeur d'un objet InetAddress doit toujours être cohérente avec la valeur de l'objet InetAddressType associé. Les tentatives de mettre un objet InetAddress à une valeur incohérente avec le InetAddressType associé doivent échouer avec une erreur inconsistentValue.

Lorsque cette convention textuelle est utilisée comme syntaxe d'un objet d'index, il peut y avoir des sorties avec la limite de 128 sous identifiants spécifiée dans SMIV2, STD 58. Dans ce cas, la définition d'objet DOIT inclure une clause 'SIZE' pour limiter le nombre d'instances potentielles de sous identifiants ; autrement, les contraintes applicables DOIVENT être déclarées dans les clauses appropriées de DESCRIPTION de rangées conceptuelles, ou dans la documentation environnante si il n'y a pas une seule clause DESCRIPTION appropriée."

SYNTAX OCTET STRING (SIZE (0..255))

InetAddressIPv4 ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1d.1d.1d.1d"

STATUT actuel

DESCRIPTION

"Représente une adresse réseau IPv4 :

Octets	Contenu	Codage
1 à 4	adresse IPv4	ordre des octets du réseau

La valeur de InetAddressType correspondante est ipv4(1).

Cette convention textuelle NE DEVRAIT PAS être utilisée directement dans les définitions d'objet, car elle restreint les adresses à un format spécifique. Cependant, si elle est utilisée, elle PEUT être utilisée soit par elle-même soit en conjonction avec InetAddressType, comme une paire."

SYNTAX OCTET STRING (SIZE (4))

InetAddressIPv6 ::= TEXTUAL-CONVENTION

DISPLAY-HINT "2x:2x:2x:2x:2x:2x:2x:2x"

STATUT actuel

DESCRIPTION

"Représente une adresse réseau IPv6 :

Octets	Contenu	Codage
1 à 16	adresse IPv6	ordre des octets du réseau

La valeur InetAddressType correspondante est ipv6(2).

Cette convention textuelle NE DEVRAIT PAS être utilisée directement dans les définitions d'objet, car elle restreint les adresses à un format spécifique. Cependant, si elle est utilisée, elle PEUT être utilisée soit par elle-même soit en conjonction avec InetAddressType, comme une paire."

SYNTAX OCTET STRING (SIZE (16))

InetAddressIPv4z ::= TEXTUAL-CONVENTION

DISPLAY-HINT "1d.1d.1d.1d%4d"

STATUT actuel

DESCRIPTION

"Représente une adresse réseau IPv4 non mondiale, avec son index de zone :

Octets	Contenu	Codage
1 à 4	adresse IPv4	ordre des octets du réseau
5 à 8	index de zone	ordre des octets du réseau

La valeur InetAddressType correspondante est ipv4z(3).

L'index de zone (octets 5 à 8) est utilisé pour ôter toute ambiguïté sur les valeurs d'adresse identiques sur les nœuds qui ont des interfaces rattachées à des zones différentes sur la même cible. L'index de zone peut contenir la valeur spéciale 0, qui se réfère à la zone par défaut pour chaque cible.

Cette convention textuelle NE DEVRAIT PAS être utilisée directement dans les définitions d'objet, car elle restreint les adresses à un format spécifique. Cependant, si elle est utilisée, elle PEUT être utilisée soit par elle-même soit en conjonction avec InetAddressType, comme une paire."SYNTAX OCTET STRING (SIZE (8))

InetAddressIPv6z ::= TEXTUAL-CONVENTION

DISPLAY-HINT "2x:2x:2x:2x:2x:2x:2x:2x%4d"

STATUT actuel

DESCRIPTION

"Représente une adresse réseau IPv6 non mondiale, avec son index de zone :

Octets	Contenu	Codage
1 à 16	adresse IPv6	ordre des octets du réseau
17 à 20	index de zone	ordre des octets du réseau

La valeur `InetAddressType` correspondante est `ipv6z(4)`.

L'index de zone (octets 17 à 20) est utilisée pour ôter toute ambiguïté sur les valeurs d'adresse identiques sur les nœuds qui ont des interfaces rattachées à des zones différentes sur la même cible. L'index de zone peut contenir la valeur spéciale 0, qui se réfère à la zone par défaut pour chaque cible.

Cette convention textuelle NE DEVRAIT PAS être utilisée directement dans les définitions d'objet, car elle restreint les adresses à un format spécifique. Cependant, si elle est utilisée, elle PEUT être utilisée soit par elle-même soit en conjonction avec `InetAddressType`, comme une paire."

SYNTAX OCTET STRING (SIZE (20))

`InetAddressDNS ::= TEXTUAL-CONVENTION`

DISPLAY-HINT "255a"

STATUT actuel

DESCRIPTION

"Représente un nom de domaine DNS. Le nom DEVRAIT être pleinement qualifié chaque fois que possible.

Le `InetAddressType` correspondant est `dns(16)`.

La clause DESCRIPTION des objets `InetAddress` qui peuvent avoir des valeurs `InetAddressDNS` DOIT décrire complètement comment (et quand) ces noms sont à résoudre en adresses IP.

La résolution d'une valeur `InetAddressDNS` peut requérir d'interroger plusieurs enregistrements DNS (par exemple, A pour IPv4 et AAAA pour IPv6). L'ordre du processus de résolution et quels enregistrement de DNS ont la priorité dépend de la configuration de l'outil de résolution.

Cette convention textuelle NE DEVRAIT PAS être utilisée directement dans les définitions d'objet, car elle restreint les adresses à un format spécifique. Cependant, si elle est utilisée, elle PEUT être utilisée soit par elle-même soit en conjonction avec `InetAddressType`, comme une paire."

SYNTAX OCTET STRING (SIZE (1..255))

`InetAddressPrefixLength ::= TEXTUAL-CONVENTION`

DISPLAY-HINT "d"

STATUT actuel

DESCRIPTION

"Note la longueur d'un préfixe d'adresse réseau Internet générique. Une valeur de n correspond à un gabarit d'adresse IP qui a n 1-bits contigus à partir du bit de plus fort poids (MSB), tous les autres bits étant mis à 0.

Une valeur de `InetAddressPrefixLength` est toujours interprétée dans le contexte d'une valeur `InetAddressType`. Tout usage de la convention textuelle `InetAddressPrefixLength` est obligé de spécifier l'objet `InetAddressType` qui fournit le contexte. Il est suggéré que l'objet `InetAddressType` soit enregistré logiquement avant le ou les objets qui utilisent la convention textuelle `InetAddressPrefixLength`, si ils apparaissent dans la même rangée logique.

Les valeurs de `InetAddressPrefixLength` plus grandes que la longueur maximum d'une adresse IP pour un `InetAddressType` spécifique sont traitées comme la valeur significative maximum applicable pour le `InetAddressType`. La valeur significative maximum est de 32 pour les `InetAddressType` 'ipv4(1)' et 'ipv4z(3)' et de 128 pour les `InetAddressType` 'ipv6(2)' et 'ipv6z(4)'. La valeur significative maximum pour le `InetAddressType` 'dns(16)' est 0.

La valeur zéro est spécifique de l'objet et doit être définie au titre de la description de tout objet qui utilise cette syntaxe. Des exemples d'usage du zéro peuvent inclure des situations où le préfixe d'adresse réseau Internet network est inconnue ou ne s'applique pas.

La limite supérieur de longueur de préfixe a été choisie pour qu'elle soit cohérente avec la taille maximum d'une `InetAddress`."

SYNTAX Unsigned32 (0..2040)

InetPortNumber ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUT actuel

DESCRIPTION

"Représente un numéro d'accès de 16 bits d'un protocole de couche transport Internet. Les numéros d'accès sont alloués par l'IANA. Une liste actuelle de toutes les allocations est disponible à <<http://www.iana.org/>>.

La valeur zéro est spécifique de l'objet et doit être définie au titre de la description de tout objet qui utilise cette syntaxe. Des exemples d'usage de zéro peuvent inclure des situations où un numéro d'accès est inconnu, ou lorsque la valeur zéro est utilisée comme caractère générique dans un filtre."

REFERENCE "STD 6 (RFC 768), STD 7 (RFC 793) et RFC 2960"

SYNTAX Unsigned32 (0..65535)

InetAutonomousSystemNumber ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUT actuel

DESCRIPTION

"Représente un numéro de système autonome qui identifie un système autonome (AS, *Autonomous System*). Un AS est un ensemble de routeurs sous une seule administration technique, qui utilise un protocole de passerelle intérieure et une métrique commune pour acheminer les paquets au sein de l'AS, et qui utilise un protocole de passerelle extérieure pour acheminer les paquets vers d'autres AS. L'IANA conserve les espaces de numéros d'AS et en a délégué une grande partie aux registres régionaux.

Les numéros de systèmes autonomes sont actuellement limités à 16 bits (0 à 65535). Il y a cependant des travaux en cours pour élargir l'espace de numéros des systèmes autonomes à 32 bits. Donc cette convention textuelle utilise une valeur Unsigned32 sans restriction de gamme afin de prendre en charge un plus grand espace de numéros de systèmes autonomes."

REFERENCE "RFC 1771, RFC 1930"

SYNTAX Unsigned32

InetScopeType ::= TEXTUAL-CONVENTION

STATUT actuel

DESCRIPTION

"Représente un type de cible. Cette convention textuelle peut être utilisée dans les cas où un MIB doit représenter différents types de cible et où il n'y a pas d'informations de contexte, comme un objet InetAddress, qui définit implicitement le type de cible.

Noter que toutes les valeurs possibles n'ont pas encore été allouées, mais elles pourront être allouées dans de futures révisions de la présente spécification. Les applications devraient donc être capables de traiter des valeurs non encore allouées."

REFERENCE "RFC 3513"

SYNTAX INTEGER {

-- reserved(0),

interfaceLocal(1),

linkLocal(2),

subnetLocal(3),

adminLocal(4),

siteLocal(5), -- Les adresses en envoi individuel de site local ont été déconseillées par la RFC 3879

-- unassigned(6),

-- unassigned(7),

organizationLocal(8), -- unassigned(9),

-- unassigned(10),

-- unassigned(11),

-- unassigned(12),

-- unassigned(13),

global(14)

-- reserved(15)

}

InetZoneIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUT actuel

DESCRIPTION

"Un index de zone identifie une instance d'une zone d'une cible spécifique.

L'index de zone DOIT ôter toute ambiguïté aux valeurs d'adresse identiques. Pour les adresses de liaison locale, l'index de zone sera normalement l'index d'interface (ifIndex est défini dans le IF-MIB) de l'interface sur laquelle l'adresse est configurée.

L'index de zone peut contenir la valeur spéciale 0, qui se réfère à la zone par défaut. La zone par défaut peut être utilisée dans les cas où l'index de zone valide n'est pas connu (par exemple, quand une application de gestion doit écrire une adresse IPv6 de liaison locale sans connaître la valeur d'index de l'interface). La zone par défaut NE DEVRAIT PAS être utilisée par facilité dans des cas où l'index de zone est connu pour une adresse IPv6 non mondiale."

REFERENCE "RFC4007"

SYNTAX Unsigned32

InetVersion ::= TEXTUAL-CONVENTION

STATUT actuel

DESCRIPTION

"Valeur représentant une version du protocole IP.

unknown(0) Version inconnue ou non spécifiée du protocole IP.

ipv4(1) Protocole IPv4 tel que défini dans la RFC 791 (STD 5).

ipv6(2) Protocole IPv6 tel que défini dans la RFC 2460.

Noter que cette convention textuelle NE DEVRAIT PAS être utilisée pour distinguer des types d'adresse différents associés aux protocoles IP. InetAddressType a été conçu à cette fin."

REFERENCE "RFC 791, RFC 2460"

SYNTAX INTEGER {

unknown(0),

pv4(1),

ipv6(2)

}

END

4 Conseils d'utilisation

Les conventions textuelles InetAddressType et InetAddress ont été introduites pour éviter des contraintes excessives dans les définitions d'objet par l'utilisation du type de base SMI IPAddress, qui est spécifique de IPv4. Une paire InetAddressType/InetAddress peut représenter des adresses IP dans divers formats.

Les objets InetAddressType et InetAddress NE DEVRAIENT PAS être subdivisés en sous-types dans les définitions d'objet. La création de sous-types lie le module MIB à des formats d'adresse spécifiques, qui peuvent causer de sérieux problèmes si de nouveaux formats d'adresse doivent être introduits. Noter qu'il est possible d'écrire des déclarations de conformité qui indiquent que seul un sous ensemble des types d'adresse définis doivent être mis en œuvre pour être conforme.

Toute utilisation des conventions textuelles InetAddress ou InetAddressPrefixLength doit spécifier quel objet InetAddressType fournit le contexte pour l'interprétation de la convention textuelle InetAddress ou InetAddressPrefixLength.

Il est suggéré que l'objet InetAddressType soit enregistré logiquement avant le ou les objets qui utilisent la convention textuelle InetAddress ou InetAddressPrefixLength. Un objet InetAddressType est enregistré logiquement avant un objet InetAddress ou InetAddressPrefixLength si il apparaît avant l'objet InetAddress ou InetAddressPrefixLength dans la rangée conceptuelle (qui inclut tous les objets d'index). Cette règle permet à des programmes comme les compilateurs de MIB d'identifier le InetAddressType d'un objet InetAddress ou InetAddressPrefixLength donné en recherchant l'objet InetAddressType, qui précède un objet InetAddress ou InetAddressPrefixLength.

4.1 Indexage de tableau

Lorsque une adresse Internet générique est utilisée comme index, les objets `InetAddressType` et `InetAddress` DOIVENT tous deux être utilisés. L'objet `InetAddressType` DOIT figurer sur la liste avant l'objet `InetAddress` dans la clause `INDEX`.

Le mot-clé `IMPLIED NE DOIT PAS` être utilisé pour un objet de type `InetAddress` dans une clause `INDEX`. Les sous-identifiants dans de telles instances sont alors de la forme `T.N.O1.O2...On`, où `T` est la valeur de l'objet `InetAddressType` object, `O1...On` sont les octets dans l'objet `InetAddress`, et `N` est le nombre de ces octets.

Il y a un ordre lexicographique significatif des tableaux indexés de cette façon. Les applications de générateur de commandes peuvent rechercher les adresses spécifiques de type et valeur connus, produire des demandes `GetNext` pour des adresses d'un seul type, ou produire des demandes `GetNext` pour un préfixe spécifique de type et d'adresse.

4.2 Unicité des adresses

Les adresses IPv4 étaient destinées à être uniques au monde, malgré l'usage courant. Les adresses IPv6 ont été structurées pour avoir des cibles différentes et donc une univocité complète [RFC3513]. En particulier, les adresses IPv6 en envoi individuel "de liaison locale" ne sont pas garanties quant à leur unicité sur un nœud particulier. Dans un tel cas, les adresses qui se dupliquent doivent être configurées sur des interfaces différentes. Ainsi, la combinaison d'une adresse IPv6 et d'un index de zone est unique [RFC4007].

La convention textuelle `InetAddressIPv6` a été définie pour représenter des adresses IPv6 mondiales et non mondiales dans les cas où il n'est pas besoin d'index de zone (par exemple, sur des hôtes d'extrémité avec une seule interface). La convention textuelle `InetAddressIPv6z` a été définie pour représenter des adresses IPv6 non mondiales dans les cas où il est besoin d'un index de zone (par exemple, un routeur qui se connecte à plusieurs zones). Donc, les concepteurs de MIB qui utilisent des paires `InetAddressType/InetAddress` n'ont pas besoin de définir des objets supplémentaires afin de prendre en charge des adresses non mondiales sur des nœuds qui connectent plusieurs zones.

`InetAddressIPv4z` est destiné à l'usage des modules MIB (tels que le TCP-MIB) qui rapportent les adresses dans la famille d'adresses utilisées sur le réseau, mais où l'entité instrumentée obtient ces adresses de la part d'applications ou d'administrateurs sous une forme qui comporte un index de zone, comme des adresses IPv6 transposées en v4.

La taille de l'index de zone a été choisie de sorte à être cohérente avec :

- l'index numérique de zone, défini dans la RFC 4007,
- le champ `sin6_scope_id` de la structure `sockaddr_in6`, définie dans la RFC 2553.

4.3 Plusieurs adresses par hôte

Un seul système hôte peut être configuré avec plusieurs adresses (IPv4 ou IPv6), et éventuellement avec plusieurs noms DNS. Il est donc possible à un seul système hôte d'être accessibles par plusieurs paires `InetAddressType/InetAddress`.

Si cela devait poser un problème de mise en œuvre ou d'utilisation, la clause `DESCRIPTION` des objets en cause devrait décrire de façon complète l'adresse qui est mentionnée dans une telle paire `InetAddressType/InetAddress`.

4.4 Résolution des noms DNS

Les noms DNS DOIVENT être résolus en adresses IP lorsqu'il est besoin de communiquer avec l'hôte en question. Cela soulève l'aspect temporel de la définition des objets de MIB dont la valeur est un nom DNS : quand le nom est-il traduit en adresse ?

Par exemple, considérons un objet défini comme indiquant une destination de transmission, et dont la valeur est un nom DNS. Quand l'entité qui transmet résout-elle le nom DNS ? Chaque fois que survient la retransmission, ou seulement une fois lorsque l'objet a été instancié ?

La clause `DESCRIPTION` de ces objets DEVRAIT définir précisément comment et quand est faite la résolution en adresse de tout nom requis.

De même, la clause DESCRIPTION de ces objets DEVRAIT définir comment et quand est faite une recherche inverse, si un agent a accédé à une instrumentation qui connaît une adresse IP, et si le module MIB ou une mise en œuvre exige qu'il transpose l'adresse IP en nom DNS.

5 Exemple d'indexage de tableau

Cet exemple montre un tableau qui fait la liste d'homologues de communication qui sont identifiés par une adresse IPv4, une adresse IPv6, ou un nom DNS. La définition du tableau interdit aussi les entrées avec une adresse vide (dont le type serait "unknown" (*inconnu*)). La taille d'un nom DNS est limitée à 64 caractères afin de satisfaire aux contraintes de longueur des OID (*identifiant d'objet*).

peerTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF PeerEntry
MAX-ACCESS  not-accessible
STATUT      actuel
DESCRIPTION
"Liste d'homologues de communication."
::= { somewhere 1 }
```

peerEntry OBJECT-TYPE

```
SYNTAX      PeerEntry
MAX-ACCESS  non accessible
STATUT      actuel
DESCRIPTION
"Entrée contenant des informations sur un homologue particulier."
INDEX { peerAddressType, peerAddress }
::= { peerTable 1 }
```

```
PeerEntry ::= SEQUENCE {
  peerAddressType InetAddressType,
  peerAddress     InetAddress,
  peerStatus      INTEGER
}
```

peerAddressType OBJECT-TYPE

```
SYNTAX      InetAddressType
MAX-ACCESS  not-accessible
STATUT      actuel
DESCRIPTION
"Type d'adresse Internet par laquelle l'homologue est joignable."
::= { peerEntry 1 }
```

peerAddress OBJECT-TYPE

```
SYNTAX      netAddress (SIZE (1..64))
MAX-ACCESS  non accessible
STATUT      actuel
DESCRIPTION
"Adresse Internet pour l'homologue. Le type de cette adresse est déterminé par la valeur de l'objet peerAddressType. Noter que les mises en œuvre doivent se limiter à une seule entrée par homologue joignable dans ce tableau. La peerAddress ne doit pas être vide à cause de la restriction SIZE. Si une rangée est créée administrativement par une opération SNMP et si la valeur du type d'adresse est dns(16), l'agent mémorise alors le nom DNS en interne. Une recherche de nom DNS doit être effectuée sur le nom DNS mémorisé en interne chaque fois qu'il est utilisé pour contacter l'homologue. Si une rangée est créée par l'entité gérée elle-même et que la valeur de type d'adresse est dns(16), l'agent mémorise alors l'adresse IP en interne. Une recherche DNS inverse doit être effectuée sur l'adresse IP mémorisée en interne chaque fois que la valeur est restituée via SNMP."
::= { peerEntry 2 }
```

La déclaration de conformité suivante spécifie que les mises en œuvre conformes ont seulement besoin de prendre en

charge les adresses IPv4/IPv6 sans index de zone. La prise en charge des noms DNS ou des adresses IPv4/IPv6 avec des index de zone n'est pas exigée.

peerCompliance MODULE-COMPLIANCE

```

STATUT      actuel
DESCRIPTION
"Déclaration de conformité du MIB d'homologues."
MODULE      -- ce module
MANDATORY-GROUPS      peerGroup }
OBJECT      peerAddressType
SYNTAX      InetAddressType { ipv4(1), ipv6(2) }
DESCRIPTION
"Il n'est exigé d'une mise en œuvre que la prise en charge des adresses IPv4 et IPv6 sans index de zone."
::= { somewhere 2 }

```

Noter que SMIV2 ne permet pas l'inclusion d'objets qui ne sont pas accessibles dans un groupe d'objets (voir au paragraphe 3.1 du STD 58, RFC 2580). Il n'est donc pas possible de préciser la syntaxe des objets auxiliaires qui ne sont pas accessibles. Il est suggéré que la précision soit exprimée de façon informelle dans la clause DESCRIPTION de la macro invocation MODULE-COMPLIANCE.

6 Considérations sur la sécurité

Ce module ne définit aucun objet de gestion. Au contraire, il définit un ensemble de conventions textuelles qui peuvent être utilisées par d'autres modules MIB pour définir des objets de gestion.

Des considérations significatives sur la sécurité ne peuvent être écrites que dans les modules MIB qui définissent des objets de gestion. Le présent document n'a donc pas d'impact sur la sécurité de l'Internet.

7 Remerciements

Le présent document a été produit par l'équipe de conception du domaine de fonctionnement et de gestion "IPv6MIB". Les auteurs tiennent à remercier Fred Baker, Randy Bush, Richard Draves, Mark Ellison, Bill Fenner, Jun-ichiro Hagino, Mike Heard, Tim Jenkins, Allison Mankin, Glenn Mansfield, Keith McCloghrie, Thomas Narten, Erik Nordmark, Peder Chr. Norgaard, Randy Presuhn, Andrew Smith, Dave Thaler, Kenneth White, Bert Wijnen, et Brian Zill pour leurs commentaires et leurs suggestions.

8 Changements de la RFC 3291 à la RFC 4001

Les changements suivants ont été apportés par rapport à la RFC 3291 :

- o Ajout d'une restriction de gamme à la convention textuelle InetAddressPrefixLength.
- o Ajout de nouvelles conventions textuelles InetZoneIndex, InetScopeType, et InetVersion.
- o Ajout des DISPLAY-HINT "d" explicites pour les conventions textuelles qui n'en avaient pas.
- o Mise à jour des paragraphes passe partout et des références.

9 Changements de la RFC 2851 à la RFC 3291

Les changements suivants ont été apportés par rapport à la RFC 2851 :

- o Ajout des nouvelles conventions textuelles InetAddressPrefixLength, InetPortNumber, et InetAutonomousSystemNumber.
- o Réécriture de l'introduction pour dire clairement que, en général, on devrait définir les tableaux de MIB qui fonctionnent avec toutes les versions d'IP. L'autre approche de plusieurs tableaux pour les différentes versions d'IP est fortement déconseillée.

- o Ajout d'un texte aux descriptions de InetAddressType et InetAddress exigeant que les mises en œuvre rejettent les opérations d'ensemble avec une valeur de inconsistentValue si elles conduisent à des incohérences.
- o Retrait des contraintes d'ordre strict. Les clauses DESCRIPTION doivent maintenant expliquer quel objet InetAddressType fournit le contexte pour un objet InetAddress ou InetAddressPrefixLength.
- o Alignement de la formulation avec le document d'architecture de cible IPv6.
- o Partage de la convention textuelle InetAddressIPv6 en deux conventions textuelles (InetAddressIPv6 et InetAddressIPv6z) et introduction d'une nouvelle convention textuelle InetAddressIPv4z. Ajout des numéros de dénominations ipv4z(3) et ipv6z(4) à l'énumération InetAddressType. Les motifs de ce changement sont : de permettre l'introduction de conventions textuelles pour les adresses IPv4 non mondiales, de s'aligner sur les conventions textuelles pour les adresses de transport, d'avoir des déclarations de conformité plus simples dans les cas où la prise en charge des adresses IPv6 avec les index de zone n'est pas exigé, et de simplifier la mise en œuvre de systèmes hôtes qui n'auront jamais à rapporter d'index de zone.

10 Références

10.1 Références normatives

- [RFC2119] Bradner, S., "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, RFC 2119, mars 1997.
- [RFC2578] McCloghrie, K., Perkins, D., et J. Schoenwaelder, "Structure des informations de gestion Version 2 (SMIv2)", STD 58, RFC 2578, avril 1999.
- [RFC2579] McCloghrie, K., Perkins, D., et J. Schoenwaelder, "Conventions textuelles pour SMIv2", STD 58, RFC 2579, avril 1999.
- [RFC2580] McCloghrie, K., Perkins, D., et J. Schoenwaelder, "Déclarations de conformité pour SMIv2", STD 58, RFC 2580, avril 1999.
- [RFC3513] Hinden, R. et S. Deering, "Architecture d'adressage pour le protocole Internet Version 6 (IPv6)", RFC 3513, avril 2003.
- [RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., et B. Zill, "Architecture d'adresse ciblée IPv6", RFC 4007, février 2005.

10.2 Références informatives

- [RFC2553] Gilligan, R., Thomson, S., Bound, J., et W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 2553, mars 1999.
- [RFC2863] McCloghrie, K. et F. Kastenholz, "The Interfaces Group MIB", RFC 2863, juin 2000.
- [RFC3410] Case, J., Mundy, R., Partain, D., et B. Stewart, "Introduction et déclaration d'applicabilité pour le cadre de gestion des normes de l'Internet", RFC 3410, décembre 2002.
- [RFC3419] Daniele, M. and J. Schoenwaelder, "Conventions textuelles pour les adresses de transport", RFC 3419, décembre 2002.

Adresse des auteurs

Michael Daniele	Brian Haberman
SyAM Software, Inc.	Johns Hopkins University Applied Physics Laboratory
1 Chestnut St, Suite 3-I	11100 Johns Hopkins Road
Nashua, NH 03060	Laurel, MD 20723-6099
USA	USA
tél : +1 603 598-9575	tél : +1-443-778-1319
mél : michael.daniele@syamsoftware.com	mél : brian@innovationslab.net

Shawn A. Routhier	Juergen Schoenwaelder
Wind River Systems, Inc.	International University Bremen
500 Wind River Way	P.O. Box 750 561
Alameda, CA 94501	28725 Bremen
USA	Germany
tél : +1 510 749-2095	tél : +49 421 200-3587
mél : shawn.routhier@windriver.com	mél : j.schoenwaelder@iu-bremen.de

Déclaration de copyright

Copyright (C) The Internet Society (2005).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à www.rfc-editor.org, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs 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.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient 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.

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'activité de soutien administratif de l'IETF (IASA, *Administrative Support Activity*).