Groupe de travail Réseau
Request for Comments : 4213
RFC rendue obsolète : 2893
Catégorie : Sur la voie de la normalisation

E. Nordmark, Sun Microsystems, Inc.
R. Gilligan, Intransa, Inc.
octobre 2005
Traduction Claude Brière de L'Isle

Mécanismes de transition de base pour les hôtes et routeurs IPv6

Résumé
Le présent document spécifie les mécanismes de compatibilité IPv4 qui peuvent être mis en œuvre par les hôtes et routeurs IPv6. Deux mécanismes sont spécifiés, la double pile *(dual stack)* et le tunnelage configuré. La double pile implique de fournir une mise en œuvre complète des deux versions du protocole Internet (IPv4 et IPv6), et le tunnelage configuré donne le moyen de porter les paquets IPv6 sur les infrastructures d'acheminement IPv4 non modifiées.

Le présent document rend obsolète la RFC 2893.

Table des Matières

1. Introduction....1
1.1 Terminologie....2
2. Fonctionnement en double couche IP....2
2.1 Configuration d'adresse....3
2.2 DNS....3
3. Mécanismes de tunnelage configurés....3
3.1 Encapsulation....4
3.2 MTU de tunnel et fragmentation....5
3.3 Limite de bond....6
3.4 Traitement des erreurs ICMPv4....6
3.5 Construction d'en-tête IPv4....7
3.6 Désencapsulation....8
3.7 Adresses de liaison locale....9
3.8 Découverte de voisin à travers les tunnels....10
4. Menace d'usurpation de l'adresse de source....10
5. Considérations sur la sécurité....10
6. Remerciements....11
7. Références....12
7.1 Références normatives....12
7.2 Référence pour information....12
8. Changements par rapport à la RFC 2893....13
Adresse des auteurs....14
Déclaration complète de droits de reproduction....14

1. Introduction

La clé pour une transition IPv6 réussie est la compatibilité avec la large base installée d'hôtes et routeurs IPv4. Maintenir la compatibilité avec IPv4 tout en déployant IPv6 va amoindrir la tâche de faire passer l'Internet à IPv6. La présente spécification définit deux mécanismes que les hôtes et routeurs IPv6 peuvent mettre en œuvre afin d'être compatibles avec les hôtes et routeurs IPv4.

Les mécanismes de ce document sont conçus pour être employés par les hôtes et routeurs IPv6 qui ont besoin d'interopérer avec les hôtes IPv4 et d'utiliser les infrastructures d'acheminement IPv4. On s'attend à ce que la plupart des nœuds dans l'Internet auront besoin d'une telle compatibilité pour longtemps, et peut-être même indéfiniment.

Les mécanismes spécifiés ici sont :
- La double couche IP (aussi appelée double pile) : technique pour fournir un soutien complet aux deux protocoles Internet -- IPv4 et IPv6 -- dans les hôtes et routeurs.
- Le tunnelage configuré de IPv6 sur IPv4 : technique pour établir des tunnels point à point par encapsulation des paquets IPv6 au sein d'en-têtes IPv4 pour les porter sur les infrastructures d'acheminement IPv4.

Les mécanismes définis ici sont destinés à être le cœur d'une "boîte à outils de transition" -- une collection croissante de techniques de mise en œuvre et que les utilisateurs peuvent employer pour faciliter la transition. Les outils peuvent être utilisés comme de besoin. Les mises en œuvre et sites décident quelles techniques sont appropriées à leurs besoins spécifiques.

Le présent document définit l'ensemble de base des mécanismes de transition, mais ce ne sont pas les seuls outils disponibles. Des mécanismes supplémentaires de transition et de compatibilité sont spécifiés dans d'autres documents.

1.1 Terminologie

Les termes suivants sont utilisées dans ce document :

Types de nœuds :

Nœud seulement IPv4 : hôte ou routeur qui met en œuvre seulement IPv4. Un nœud seulement IPv4 ne comprend pas IPv6. La base installée d'hôtes et routeurs IPv4 existants avant que commence la transition est constituée de nœuds seulement IPv4.

Nœud IPv6/IPv4 : hôte ou routeur qui met en œuvre IPv4 et IPv6.

Nœud IPv6 seulement : hôte ou routeur qui met en œuvre IPv6 et ne met pas en œuvre IPv4. Le fonctionnement de nœuds seulement IPv6 n'est pas traité dans le présent mémoire.

Nœud IPv6 : tout hôte ou routeur qui met en œuvre IPv6. Les nœuds IPv6/IPv4 et IPv6 seulement sont tous des nœuds IPv6.

Nœud IPv4 : tout hôte ou routeur qui met en œuvre IPv4. Les nœuds IPv6/IPv4 et IPv4 seulement sont tous des nœuds IPv4.

Techniques utilisées dans la transition :

tunnelage IPv6 sur IPv4: technique d'encapsulation des paquets IPv6 au sein de IPv4 afin qu'ils puissent être portés à travers les infrastructures d'acheminement IPv4.

tunnelage configuré : tunnelage IPv6 sur IPv4 où la ou les adresses de point d'extrémité de tunnel IPv4 sont déterminées par les informations de configuration sur les points d'extrémité de tunnel. Tous les tunnels sont supposés être bidirectionnels. Le tunnel fournit une liaison (virtuelle) point à point à la couche IPv6, en utilisant les adresses IPv4 configurées comme adresses de point d'extrémité de couche inférieure.

Les autres mécanismes de transition, incluant d'autres mécanismes de tunnelage, sortent du domaine d'application du présent document.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Fonctionnement en double couche IP

La façon la plus directe pour que les nœuds IPv6 restent compatibles avec les nœuds seulement IPv4 est de fournir une mise en œuvre IPv4 complète. Les nœuds IPv6 qui fournissent une mise en œuvre IPv4 et IPv6 complète sont appelés des

"nœuds IPv6/IPv4". Les nœuds IPv6/IPv4 ont la capacité d'envoyer et recevoir les deux types de paquets IPv4 et IPv6. Ils peuvent directement interopérer avec les nœuds IPv4 en utilisant les paquets IPv4, et aussi interopérer directement avec les nœuds IPv6 en utilisant les paquets IPv6.

Bien qu'un nœud puisse être équipé pour prendre en charge les deux protocoles, une ou l'autre pile peut être désactivée pour des raisons de fonctionnement. On utilise ici la notion assez vague de "pile". Une pile étant activée lorsque des adresses IPv4 sont allouées, mais il n'est pas explicitement défini que des applications particulières soient disponibles sur les piles. Donc, les nœuds IPv6/IPv4 peuvent fonctionner dans un des trois modes :
- avec leur pile IPv4 activée et leur pile IPv6 désactivée.
- avec leur pile IPv6 activée et leur pile IPv4 désactivée.
- avec les deux piles activées.

Les nœuds IPv6/IPv4 avec leur pile IPv6 désactivée vont fonctionner comme des nœuds seulement IPv4. De même, les nœuds IPv6/IPv4 avec leur pile IPv4 désactivée vont fonctionner comme des nœuds seulement IPv6. Les nœuds IPv6/IPv4 PEUVENT fournir un commutateur de configuration pour désactiver leur pile IPv4 ou IPv6.

La technique de tunnelage configuré, qui est décrite à la Section 3, peut ou non être utilisée en plus du fonctionnement de double couche IP.

2.1 Configuration d'adresse

Comme les nœuds prennent en charge les deux protocoles, les nœuds IPv6/IPv4 peuvent être configurés avec des adresses IPv4 comme IPv6. Les nœuds IPv6/IPv4 utilisent les mécanismes IPv4 (par exemple, DHCP) pour acquérir leurs adresses IPv4, et les mécanismes de protocole IPv6 (par exemple, l'autoconfiguration d'adresse sans état [RFC2462] et/ou DHCPv6) pour acquérir leurs IPv6.

2.2 DNS

Le système des noms de domaines (DNS) est utilisé aussi bien dans IPv4 que IPv6 pour transposer entre noms d'hôtes et adresses IP. Un nouveau type d'enregistrement de ressource appelé "AAAA" a été défini pour les adresses IPv6 [RFC3596]. Comme les nœuds IPv6/IPv4 doivent être capables d'interopérer directement avec les deux types de nœuds IPv4 et IPv6, ils doivent fournir des bibliothèques de résolveur capables de traiter les enregistrements IPv4 "A" aussi bien que les enregistrements IPv6 "AAAA". Noter que la recherche d'enregistrements A par rapport à celle sur les enregistrements AAAA est indépendante de la question de savoir si les paquets de DNS sont portés dans des paquets IPv4 ou IPv6 et qu'on ne fait pas d'hypothèse sur la question de savoir si les serveurs DNS connaissent les capacités IPv4/IPv6 du nœud demandeur.

Les questions et les lignes directrices de fonctionnement pour l'utilisation de IPv6 avec le DNS sont décrites plus en détails dans d'autres documents, par exemple, la [RFC4472].

Les bibliothèques de résolveur DNS sur les nœuds IPv6/IPv4 DOIVENT être capables de traiter les enregistrements AAAA et A. Cependant, quand une interrogation localise un enregistrement AAAA détenant une adresse IPv6, et un enregistrement A contenant une adresse IPv4, la bibliothèque de résolveur PEUT ordonner les résultats retournés à l'application dans un ordre qui influence la version de paquets IP utilisée pour communiquer avec ce nœud -- IPv6 d'abord ou IPv4 d'abord.

Les applications DEVRAIENT être capables de spécifier si elles veulent les enregistrements IPv4, IPv6, ou les deux [RFC3493]. Cela définit quelles familles d'adresses cherche le résolveur. Si il n'y a pas un choix de l'application, ou si l'application a demandé les deux, la bibliothèque de résolveur NE DOIT PAS éliminer d'enregistrements.

Comme la plupart des applications essayent les adresses dans l'ordre où le résolveur les a retournées, cela peut affecter la "préférence" de version IP des applications.

Les mécanismes réels d'ordre sortent du domaine d'application du présent mémoire. Le choix des adresses est décrit plus en détails dans la [RFC3484].

3. Mécanismes de tunnelage configurés

Dans la plupart des scénarios de déploiement, l'infrastructure d'acheminement IPv6 va être construite progressivement. Pendant que l'infrastructure IPv6 se déploie, l'infrastructure d'acheminement IPv4 existante peut rester fonctionnelle et peut être utilisée à porter du trafic IPv6. Le tunnelage donne un moyen pour utiliser une infrastructure d'acheminement IPv4 existante pour porter le trafic IPv6.

Les hôtes et routeurs IPv6/IPv4 peuvent tunneler les datagrammes IPv6 sur les régions à topologie d'acheminement IPv4 en les encapsulant au sein de paquets IPv4. Le tunnelage peut être utilisé de différentes façons :
- De routeur à routeur : les routeurs IPv6/IPv4 interconnectés par une infrastructure IPv4 peuvent tunneler les paquets IPv6 entre eux. Dans ce cas, le tunnel s'étend sur un segment du chemin de bout en bout que prend le paquet IPv6.
- D'hôte à routeur : les hôtes IPv6/IPv4 peuvent tunneler les paquets IPv6 à un routeur IPv6/IPv4 intermédiaire qui est accessible via une infrastructure IPv4. Ce type de tunnel s'étend sur le premier segment du chemin de bout en bout du paquet.
- D'hôte à hôte : les hôtes IPv6/IPv4 qui sont interconnectés par une infrastructure IPv4 peuvent tunneler les paquets IPv6 entre eux. Dans ce cas, le tunnel s'étend sur le chemin de bout en bout entier que prend le paquet.
- De routeur à hôte : les routeurs IPv6/IPv4 peuvent tunneler les paquets IPv6 à leurs hôtes IPv6/IPv4 de destination finale. Ce tunnel s'étend seulement sur le dernier segment du chemin de bout en bout.

Le tunnelage configuré peut être utilisé dans tous les cas ci-dessus, mais il est très probable qu'il sera utilisé de routeur à routeur à cause du besoin de configurer explicitement les points d'extrémité du tunnelage.

Les mécanismes sous-jacents au tunnelage sont :
- Le nœud d'entrée du tunnel (l'encapsuleur) crée un en-tête IPv4 d'encapsulation et transmet les paquets encapsulés.
- Le nœud de sortie du tunnel (le décapsuleur) reçoit le paquet encapsulé ; il réassemble le paquet si nécessaire, retire l'en-tête IPv4, et traite le paquet IPv6 reçu.
- L'encapsuleur peut avoir besoin de conserver des informations d'état mou pour chaque tunnel enregistrant des paramètres comme la MTU du tunnel afin de traiter les paquets IPv6 transmis dans le tunnel.

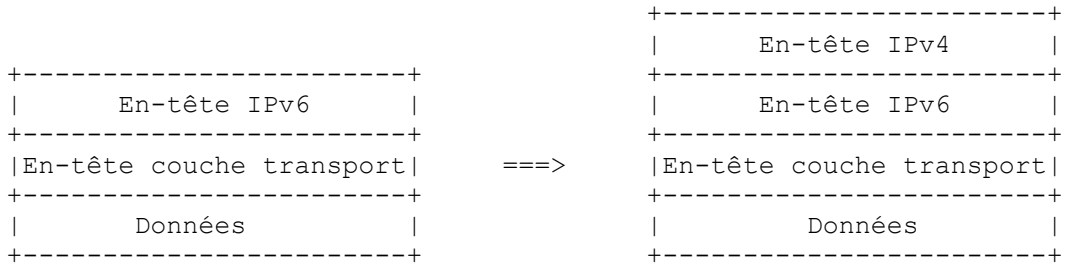
Dans le tunnelage configuré, les adresses de point d'extrémité de tunnel sont déterminées chez l'encapsuleur à partir des informations de configuration mémorisées pour chaque tunnel. Lorsque un paquet IPv6 est transmis à travers un tunnel, les adresses de destination et de source pour l'en-tête IPv4 encapsulant sont réglées comme décrit au paragraphe 3.5.

La détermination de quels paquets tunneler est généralement faite par les informations d'acheminement à l'encapsuleur. C'est généralement fait via un tableau d'acheminement, qui dirige les paquets sur la base de leur adresse de destination en utilisant le gabarit de préfixe et la technique de confrontation.

Le décapsuleur confronte les paquets de protocole 41 reçus sur les tunnels qu'il a configuré, et n'admet que les paquets dans lesquels les adresses de source IPv4 correspondent aux tunnels configurés chez le décapsuleur. Donc, l'opérateur doit s'assurer que la configuration de l'adresse IPv4 du tunnel est la même chez l'encapsuleur et chez le décapsuleur.

3.1 Encapsulation

La Figure ci-dessous montre l'encapsulation d'un datagramme IPv6 dans IPv4 :



Encapsulation d'IPv6 dans IPv4

En plus de l'ajout d'un en-tête IPv4, l'encapsuleur a aussi à traiter des questions plus complexes :
- Déterminer quand fragmenter et quand rapporter une erreur ICMPv6 "Paquet trop gros" à la source.
- Comment refléter à la source les erreurs ICMPv4 à partir des routeurs le long du tunnel comme des erreurs ICMPv6.

Ces questions sont discutées dans les paragraphes qui suivent.

3.2 MTU de tunnel et fragmentation

Naïvement, l'encapsuleur pourrait voir l'encapsulation IPv6 utilisant IPv4 comme une couche de liaison avec une très grande MTU (65535-20 octets au plus ; 20 octets "supplémentaires" sont nécessaires pour l'en-tête d'encapsulation IPv4). L'encapsuleur aurait seulement besoin de rapporter les erreurs ICMPv6 "Paquet trop gros" à la source pour les paquets qui excèdent cette MTU. Cependant, un tel schéma serait inefficace ou non interopérable pour trois raisons et donc NE DOIT PAS être utilisé :

1) Il en résulterait plus de fragmentations que nécessaire. Les fragmentation à la couche IPv4 devraient être évitées à cause des problèmes de performances causés parce l'unité de perte serait plus petite que l'unité de retransmission [KM97].

2) Toute fragmentation IPv4 se produisant à l'intérieur du tunnel, c'est-à-dire, entre l'encapsuleur et le décapsuleur, va devoir être réassemblée au point d'extrémité du tunnel. Pour les tunnels qui se terminent à un routeur, cela exigerait de la mémoire supplémentaire et d'autres ressources pour rassembler les fragments IPv4 en un paquet IPv6 complet avant que le paquet puisse être transmis.

3) L'encapsuleur n'a pas de moyen de savoir si le décapsuleur est capable de défragmenter de tels paquets IPv4 (voir les détails au paragraphe 3.6) et n'a aucun moyen de savoir si le décapsuleur est capable de traiter une unité de réception maximum (MRU, *Maximum Receive Unit*) IPv6 aussi grande.

Donc, l'encapsuleur NE DOIT PAS traiter le tunnel comme une interface avec une MTU de 64 kilooctets, mais plutôt utiliser la MTU fixe statique ou la détermination FACULTATIVE de MTU dynamique sur la base de la MTU de chemin IPv4 au point d'extrémité du tunnel.

Si les deux mécanismes sont mis en œuvre, la décision duquel utiliser DEVRAIT être configurable en fonction du point d'extrémité de tunnel.

3.2.1 MTU statique de tunnel

Un nœud qui utilise la MTU statique de tunnel traite l'interface de tunnel comme ayant une MTU fixe d'interface. Par défaut, la MTU DOIT être entre 1280 et 1480 octets (inclus) mais DEVRAIT être de 1280 octets. Si la valeur par défaut n'est pas 1280 octets, la mise en œuvre DOIT avoir un bouton de réglage de configuration qui peut être utilisé pour changer la valeur de la MTU.

Un nœud doit être capable d'accepter un paquet IPv6 fragmenté qui, après réassemblage, fait 1500 octets [RFC2460]. Le présent mémoire comporte aussi des exigences (voir au paragraphe 3.6) sur la quantité de réassemblage IPv4 et de MRU IPv6 qui DOIVENT être acceptées par tous les décapsuleurs. Cela assure une interopérabilité correcte avec toutes les MTU fixes entre 1280 et 1480 octets.

Une MTU fixe plus grande que ce qui est accepté par ces exigences ne doit pas être configurée sauf si on s'est administrativement assuré que le décapsuleur peut réassembler ou recevoir des paquets de cette taille.

Le choix d'une bonne MTU de tunnel dépend de nombreux facteurs, au moins :
- Si les paquets de protocole-41 IPv4 vont être transportés sur des supports qui peuvent avoir une MTU de chemin inférieure (par exemple, des réseaux privés virtuels IPv4) ; prendre une valeur trop élevée peut alors conduire à la fragmentation IPv4.
- Si le tunnel est utilisé pour transporter des paquets IPv6 tunnelés (par exemple, un nœud mobile avec un tunnel configuré IPv6 dans IPv4, et une interface de tunnel IPv6 dans IPv6) ; prendre alors une valeur trop faible peut conduire à la fragmentation IPv6.

Si l'encapsulation en couche est estimée être présente, il peut être prudent d'envisager de prendre plutôt en charge la détermination de la MTU dynamique car elle est capable de minimiser la fragmentation et optimise les tailles de paquet.

Lorsque on utilise la MTU statique de tunnel, le bit Ne Pas Fragmenter NE DOIT PAS être établi dans l'en-tête d'encapsulation IPv4. Par suite, l'encapsuleur ne devrait pas recevoir de message ICMPv4 "Paquet trop gros" en résultat des paquets qu'il a encapsulé.

3.2.2 MTU dynamique de tunnel

La détermination dynamique de la MTU est FACULTATIVE. Cependant, si elle est mise en œuvre, elle DEVRAIT avoir le comportement décrit dans le présent document.

La fragmentation à l'intérieur du tunnel peut être réduite au minimum en faisant que l'encapsuleur retrace la MTU de chemin IPv4 à travers le tunnel, en utilisant le protocole de découverte de la MTU de chemin IPv4 [RFC1191] et en enregistrant la MTU de chemin résultante. La couche IPv6 chez l'encapsuleur peut alors voir un tunnel comme une couche de liaison avec une MTU égale à la MTU de chemin IPv4, moins la taille de l'en-tête IPv4 encapsulant.

Noter que cela n'élimine pas la fragmentation IPv4 dans le cas où la MTU de chemin IPv4 résulterait en une MTU IPv6 inférieure à 1280 octets. (Toute couche de liaison utilisée par IPv6 doit avoir une MTU d'au moins 1280 octets [RFC2460].) Dans ce cas, la couche IPv6 doit "voir" une couche de liaison avec une MTU de 1280 octets et l'encapsuleur doit utiliser la fragmentation IPv4 afin de transmettre les paquets IPv6 de 1280 octets.

L'encapsuleur DEVRAIT employer l'algorithme suivant pour déterminer quand transmettre un paquet IPv6 qui fait plus que la MTU de chemin du tunnel en utilisant la fragmentation IPv4, et quand retourner un message ICMPv6 "Paquet trop gros" selon la [RFC1981] :

```
si (MTU de chemin IPv4 - 20) est inférieure à 1280
    si le paquet fait plus de 1280 octets
        Envoyer ICMPv6 "Paquet trop gros" avec MTU = 1280.
        Éliminer le paquet.
    autrement
        Encapsuler sans établir le fanion Ne Pas Fragmenter dans l'en-tête IPv4.
        Le paquet IPv4 résultant peut être fragmenté par la couche IPv4 chez l'encapsuleur
        ou par un routeur le long du chemin IPv4.
        fin de si
    autrement
      si le paquet fait plus que (MTU de chemin IPv4 - 20)
          Envoyer ICMPv6 "Paquet trop gros" avec MTU = (MTU de chemin IPv4 - 20).
          Éliminer le paquet.
    autrement
        Encapsuler et établir le fanion Ne Pas Fragmenter dans l'en-tête IPv4.
    fin de si
fin de si
```

Les encapsuleurs qui ont un grand nombre de tunnels peuvent choisir entre MTU de tunnel dynamique ou statique selon le point d'extrémité de tunnel. Dans les cas où le nombre de tunnels qu'un nœud utilise est grand, il est utile d'observer que ces informations d'état peuvent être mises en antémémoire et éliminées quand on en a plus l'usage.

Noter que l'utilisation de la MTU de tunnel dynamique est sujette aux trous noirs de la MTU de chemin IPv4 si les messages ICMPv4 "Paquet trop gros" sont éliminés par des pare-feu ou ne sont pas générés par les routeurs [RFC1435], [RFC2923].

3.3 Limite de bond

Les tunnels IPv6 sur IPv4 sont modélisés comme un "seul bond" du point de vue de IPv6. Le tunnel est opaque aux utilisateurs du réseau, et ils ne sont pas détectables par les outils de diagnostic du réseau comme traceroute.

Le modèle à un seul bond est mis en œuvre en ayant les encapsuleurs et décapsuleurs qui traitent le champ limite de bonds IPv6 comme ils le feraient si ils transmettaient un paquet à n'importe quelle autre liaison de données. C'est-à-dire qu'ils décrémentent la limite de bonds de 1 quand ils transmettent un paquet IPv6. (Le nœud d'origine et la destination finale ne décrémentent pas la limite de bonds.)

Le TTL de l'en-tête IPv4 encapsulant est choisi de façon dépendante de la mise en œuvre. La valeur courante suggérée est publiée dans les "Numéros alloués" [ASSIGNED]. Les mises en œuvre PEUVENT fournir un mécanisme pour permettre à leur administrateur de configurer le TTL IPv4 selon la MIB de tunnel IP [RFC4087].

3.4 Traitement des erreurs ICMPv4

En réponse aux paquets encapsulés qu'il a envoyé dans le tunnel, l'encapsuleur peut recevoir des messages d'erreur ICMPv4 de routeurs IPv4 de l'intérieur du tunnel. Ces paquets sont adressés à l'encapsuleur parce que c'est la source IPv4 du paquet encapsulé.

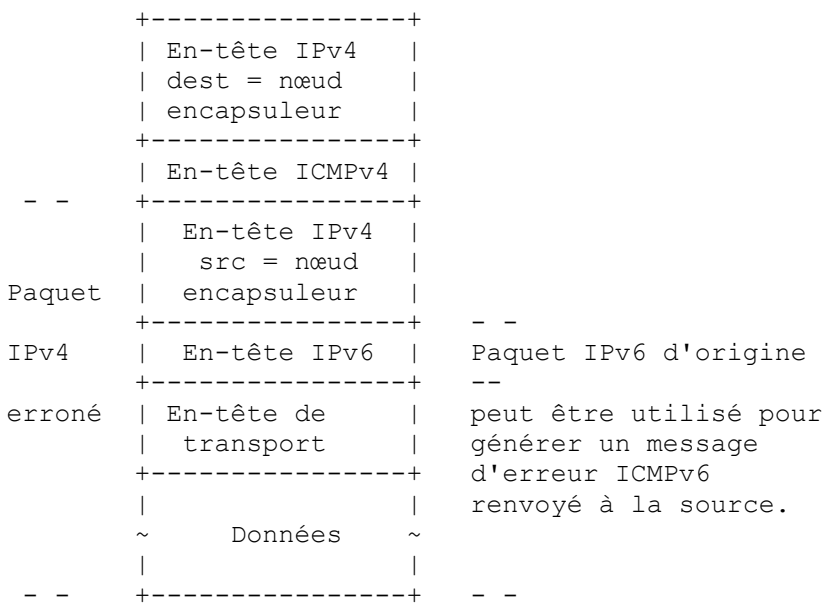
Le traitement d'erreur ICMPv4 n'est applicable qu'à la détermination de MTU dynamique, même si les fonctions pourraient aussi être utilisées avec les tunnels à MTU statique.

Les messages d'erreur ICMPv4 "Paquet trop gros" sont traités conformément à la découverte de la MTU de chemin IPv4 [RFC1191] et la MTU de chemin résultante est enregistrée dans la couche IPv4. La MTU de chemin enregistrée est utilisée par IPv6 pour déterminer si une erreur ICMPv6 "Paquet trop gros" doit être générée comme décrit au paragraphe 3.2.2.

Le traitement des autres types de message d'erreur ICMPv4 dépend de la façon dont les informations sont disponibles à partir du paquet encapsulé qui a causé l'erreur.

De nombreux routeurs IPv4 plus anciens ne retournent que 8 octets de données au delà de l'en-tête IPv4 du paquet erroné, ce qui n'est pas assez pour inclure les champs d'adresse de l'en-tête IPv6. Les routeurs IPv4 plus modernes vont probablement retourner assez de données au delà de l'en-tête IPv4 pour inclure l'en-tête IPv6 entier et éventuellement même les données au delà. Voir la [RFC1812].

Si suffisamment d'octets de données du paquet en cause sont disponibles, l'encapsuleur PEUT extraire le paquet IPv6 encapsulé et l'utiliser pour générer un message ICMPv6 dirigé sur le nœud IPv6 d'origine, comme montré ci-dessous :



Message d'erreur ICMPv4 retourné au nœud encapsulant

Lorsque on reçoit des erreurs ICMPv4 comme ci-dessus et que l'erreur n'est pas "Paquet trop gros", il serait utile d'enregistrer l'erreur comme relative au tunnel. Aussi, si des en-têtes suffisants sont disponibles, le nœud d'origine PEUT envoyer une erreur ICMPv6 de type "non accessible" avec le code "adresse non joignable" à la source IPv6. (Le code "adresse non joignable" est approprié car, du point de vue de IPv6, le tunnel est une liaison et ce code est utilisé pour les erreurs spécifiques de liaisons [RFC2463]).

Noter que quand la MTU de chemin IPv4 est dépassée, et qu'un nombre suffisant d'octets de charge utile associés à l'erreur ICMPv4 n'est pas disponible, ou que des erreurs ICMPv4 n'ont pas causé la génération des erreurs ICMPv6 dans le cas où il y a assez de charge utile, il va y avoir au moins deux paquets éliminés au lieu de au moins un (cas d'une seule couche de découverte de MTU). Considérons un cas où un hôte IPv6 est connecté à un routeur IPv4/IPv6, qui est connecté à un réseau où une erreur ICMPv4 est générée sur une taille de paquet trop grosse. D'abord, le routeur a besoin de savoir la MTU du tunnel (IPv4) qui cause la perte d'au moins un paquet, et ensuite l'hôte doit apprendre la MTU (IPv6) à partir du routeur qui cause la perte d'au moins un paquet. Il reste que dans tous les cas il peut y avoir plus d'une perte de paquet si il y a plusieurs gros paquets en cours au même moment.

3.5 Construction d'en-tête IPv4

Lorsque on encapsule un paquet IPv6 dans un datagramme IPv4, les champs de l'en-tête IPv4 sont réglés comme suit :

Version : 4

Longueur d'en-tête IP, en mots de 32 bits : 5 (il n'y a pas d'options IPv4 dans l'en-tête encapsulant.)

Type de service : 0 sauf spécification contraire. (Voir la [RFC2983] et le paragraphe 9.1 de la [RFC3168] sur les questions relatives à l'octet Type de service et le tunnelage.)

Longueur totale : Longueur de la charge utile provenant de l'en-tête IPv6 plus longueur des en-têtes IPv6 et IPv4 (c'est-à-dire, longueur de charge utile IPv6 plus une constante de 60 octets).

Identification : Généré de façon univoque comme pour tout paquet IPv4 transmis par le système.

Fanions : Établir le fanion Ne pas fragmenter (DF) comme spécifié au paragraphe 3.2. Établir le fanion Fragments à suivre (MF) comme nécessaire si il y a fragmentation.

Décalage de fragment : réglé comme nécessaire si il y a fragmentation.

Durée de vie (TTL) : réglé de façon spécifique de la mise en œuvre, comme décrit au paragraphe 3.3.

Protocole : 41 (numéro de type de charge utile alloué pour IPv6).

Somme de contrôle d'en-tête : Calculer la somme de contrôle de l'en-tête IPv4 [RFC0791].

Adresse de source : adresse IPv4 de l'encapsuleur : configurée par l'administrateur ou adresse de l'interface sortante.

Adresse de destination : adresse IPv4 du point d'extrémité du tunnel.

Lors de l'encapsulation des paquets, le nœud doit s'assurer qu'il va utiliser l'adresse de source correcte afin que les paquets soient acceptables pour le décapsuleur comme décrit au paragraphe 3.6. Configurer l'adresse de source est approprié particulièrement dans les cas dans lesquels la sélection automatique de l'adresse de source peut produire des résultats différents à certaines heures. C'est souvent le cas avec des adresses multiples, et des interfaces multiples, ou quand les chemins peuvent changer fréquemment. Donc, il DEVRAIT être possible de spécifier administrativement l'adresse de source d'un tunnel.

3.6 Désencapsulation

Lorsque un hôte ou routeur IPv6/IPv4 reçoit un datagramme IPv4 qui est adressé à une de ses propres adresses IPv4 ou à une adresse de groupe de diffusion groupée à laquelle il s'est joint, et que la valeur du champ Protocole est 41, le paquet est potentiellement un paquet tunnelé dont il est besoin de vérifier qu'il appartient à une des interfaces de tunnel configuré (en vérifiant les adresses de source/destination) de réassembler (si il est fragmenté au niveau IPv4) et de retirer l'en-tête IPv4 et de soumettre le datagramme IPv6 résultant au code de couche IPv6 sur le nœud.

Le décapsuleur DOIT vérifier que l'adresse de source du tunnel est correcte avant de poursuivre le traitement des paquets, pour atténuer les problèmes d'usurpation d'adresse (voir la Section 4). Cette vérification s'applique aussi aux paquets qui sont livrés aux protocoles de transport au décapsuleur. Cela se fait en vérifiant que l'adresse de source est l'adresse IPv4 de l'encapsuleur, telle que configurée au décapsuleur. Les paquets pour lesquels l'adresse IPv4 de source ne correspond pas DOIVENT être éliminés et un message ICMP NE DEVRAIT PAS être généré ; cependant, si la mise en œuvre envoie normalement un message ICMP quand elle reçoit un paquet de protocole inconnu, un tel message d'erreur PEUT être envoyé (par exemple, ICMPv4 Protocole 41 injoignable).

Un effet collatéral de cette vérification d'adresse est que le nœud va éliminer en silence les paquets qui ont une mauvaise adresse de source et les paquets qui ont été reçus par le nœud mais qui ne lui sont pas adressés directement (par exemple, des adresses de diffusion).

Indépendamment de toutes les autres formes de filtrage d'entrée IPv4 que l'administrateur du nœud peut avoir configurées, la mise en œuvre PEUT effectuer un filtrage d'entrée, c'est-à-dire, vérifier que le paquet arrive de l'interface dans la direction vers le point d'extrémité du tunnel, comme une vérification de transmission stricte sur le chemin inverse (RPF, *Reverse Path Forwarding*) [RFC3704]. Comme cela peut causer des problèmes sur les tunnels qui sont acheminés à travers

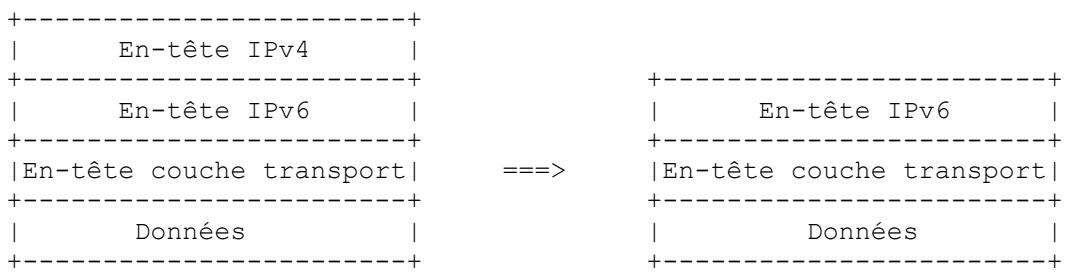
plusieurs liaisons, il est RECOMMANDÉ que cette vérification, si elle est faite, soit désactivée par défaut. Les paquets capturés par cette vérification DEVRAIENT être éliminés ; un message ICMP NE DEVRAIT PAS être généré par défaut.

Le décapsuleur DOIT être capable d'avoir, sur les interfaces du tunnel, une MRU IPv6 d'au moins le maximum de 1500 octets et la plus grande MRU d'interface (IPv6) sur le décapsuleur.

Le décapsuleur DOIT être capable de réassembler un paquet IPv4 qui est (après réassemblage) d'un maximum de 1500 octets et de la plus grande MTU d'interface (IPv4) sur le décapsuleur. Le chiffre de 1500 octets est le résultat des encapsuleurs qui utilisent le schéma de MTU statique du paragraphe 3.2.1, tandis que les encapsuleurs qui utilisent le schéma dynamique du paragraphe 3.2.2 peuvent causer la réception sur le décapsuleur jusqu'à la plus grande MTU d'interface. (Noter que c'est strictement la MTU d'interface sur le dernier routeur IPv4 *avant* le décapsuleur en cause, mais pour la plupart des liaisons la MTU est la même entre tous les voisins.)

Cette limite de réassemblage permet que la détermination de la MTU dynamique de tunnel par l'encapsuleur tire parti de plus grandes MTU de chemin IPv4. Une mise en œuvre PEUT avoir un bouton de configuration qui sera utilisé pour régler une plus grande valeur des mémoires tampon de réassemblage de tunnel que le chiffre mentionné ci-dessus, mais il NE DOIT PAS être réglé en dessous de cette valeur.

La décapsulation est montrée ci dessous :



Désencapsulation de IPv6 de IPv4

Le décapsuleur effectue le réassemblage IPv4 avant de décapsuler le paquet IPv6.

Lors de la décapsulation du paquet, l'en-tête IPv6 n'est pas modifié. (Cependant, voir la [RFC2983] et le paragraphe 9.1 de la [RFC3168] sur les questions relatives à l'octet Type de service et le tunnelage.) Si le paquet est ensuite transmis, sa limite de bonds est décrémentée de un.

L'en-tête IPv4 encapsulant est éliminé, et la validité du paquet résultant est vérifiée lorsque il est soumis à la couche IPv6. Lors de la reconstruction du paquet IPv6, la longueur DOIT être déterminée à partir de la longueur de la charge utile IPv6 car le paquet IPv4 pourrait avoir un bourrage (donc avoir une longueur supérieure à celle du paquet IPv6 plus l'en-tête IPv4 retiré).

Après la décapsulation, le nœud DOIT éliminer en silence un paquet dont l'adresse IPv4 de source est invalide. La liste des adresses de source invalides DEVRAIT inclure au moins :
- toutes les adresses de diffusion groupée (FF00::/8) ;
- l'adresse de rebouclage (::1) ;
- toutes les adresses IPv6 compatibles IPv4 [RFC3513] (::/96), à l'exclusion de l'adresse inspécifiée pour la détection d'adresse dupliquée (::/128) ;
- toutes les adresses IPv6 transposées en adresses IPv4 (::ffff:0:0/96).

De plus, le nœud devrait être configuré à effectuer le filtrage d'entrée [RFC2827], [RFC3704] sur l'adresse IPv6 de source, similaire sur toutes ses interfaces, par exemple :
1) si le tunnel est vers l'Internet, le nœud devrait être configuré à vérifier que les préfixes IPv6 du site ne sont pas utilisés comme adresses de source, ou
2) si le tunnel est vers un réseau bordure, le nœud devrait être configuré à vérifier que l'adresse de source appartient à ce réseau bordure.

La liste des préfixes doit normalement être configurée manuellement ; l'appartenance au réseau bordure peut être vérifiée automatiquement, par exemple, en utilisant une vérification de RPF stricte en envoi individuel, pour autant qu'une interface puisse être destinée à être vers une bordure.

Il est RECOMMANDÉ que les mises en œuvre fournissent un seul bouton pour faciliter un strict filtrage d'entrée par les administrateurs vers les réseaux de bordure.

3.7 Adresses de liaison locale

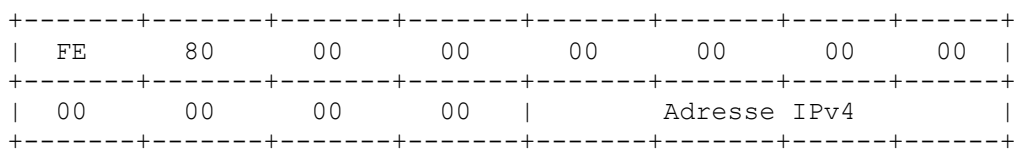
Les tunnels configurés sont des interfaces IPv6 (sur la "couche de liaison" IPv4) et DOIVENT donc avoir des adresses locales de liaison. Les adresses locales de liaison sont utilisées, par exemple, par les protocoles d'acheminement qui fonctionnent sur les tunnels.

L'identifiant d'interface [RFC3513] pour une telle interface peut être fondé sur l'adresse IPv4 de 32 bits d'une interface sous-jacente ou formée en utilisant d'autres moyens, pour autant qu'il soit unique avec une probabilité raisonnablement élevée par rapport à l'autre point d'extrémité du tunnel.

Noter qu'il peut être souhaitable de former l'adresse locale de liaison d'une façon qui minimise la probabilité, et les conséquences, d'avoir à renuméroter l'adresse locale de liaison dans l'éventualité d'un changement de topologie ou de matériel.

Si une adresse IPv4 est utilisée pour former l'adresse IPv6 locale de liaison, l'identifiant d'interface est l'adresse IPv4, précédée par des zéros. Noter que le bit "Universel/Local" est zéro, indiquant que l'identifiant d'interface n'est pas unique au monde. L'adresse locale de liaison est formée en ajoutant l'identifiant d'interface au préfixe FE80::/64.

Lorsque l'hôte a plus d'une adresse IPv4 utilisée sur l'interface physique concerné, le choix d'une de ces adresses IPv4 est fait par l'administrateur ou la mise en œuvre lors de la formation de l'adresse locale de liaison.



3.8 Découverte de voisin à travers les tunnels

Les mises en œuvre de tunnel configurées DOIVENT au moins accepter et répondre aux paquets de sondage utilisés par la détection d'inaccessibilité de voisin (NUD, *Neighbor Unreachability Detection*) [RFC2461]. Les mises en œuvre DEVRAIENT aussi envoyer des paquets de sondage NUD pour détecter quand le tunnel configuré a une défaillance pour que la mise en œuvre puisse utiliser un chemin de remplacement pour atteindre la destination. Noter que la découverte de voisin permet que l'envoi de sondes NUD soit omis pour les liaisons de routeur à routeur si le protocole d'acheminement trace l'accessibilité dans les deux directions.

Pour les besoins de la découverte de voisin, les tunnels configurés spécifiés dans ce document sont supposés N'AVOIR PAS d'adresse de couche de liaison, même si la couche de liaison (IPv4) a bien une adresse. Cela signifie que :
- l'envoyeur des paquets de découverte de voisin NE DEVRAIT PAS inclure les options Adresse de source de couche de liaison ou Adresse cible de couche de liaison dans la liaison tunnel ;
- le receveur DOIT, tout en traitant par ailleurs le paquet de découverte de voisin, ignorer en silence le contenu de toute option reçue sur la liaison tunnel.

Ne pas utiliser d'options d'adresse de couche de liaison est cohérent avec la façon dont la découverte de voisin est utilisée sur les autres liaisons en point à point.

4. Menace d'usurpation de l'adresse de source

Les spécifications ci-dessus contiennent des règles qui s'appliquent à la vérification d'adresse de source de tunnel en particulier et au filtrage d'entrée [RFC2827][RFC3704] en général aux paquets avant qu'ils soient décapsulés. Lorsque le tunnelage IP dans IP (indépendant des versions IP) est utilisé, il est important que ce ne soit pas utilisé pour contourner un filtrage d'entrée utilisé pour les paquets non tunnelés. Donc, les règles du présent document sont déduites sur la base d'un filtrage d'entrée qui devrait être utilisé pour IPv4 et IPv6, et l'utilisation du tunnelage ne devrait pas fournir un moyen facile pour contourner le filtrage.

Dans ce cas, sans vérifications spécifiques de filtrage d'entrée dans le décapsuleur, il devrait être possible à un attaquant d'injecter un paquet avec :

- une source IPv4 externe : l'adresse IPv4 réelle de l'attaquant,
- une destination IPv4 externe : l'adresse IPv4 du décapsuleur,
- une source IPv6 interne : Alice, qui est soit le décapsuleur, soit un nœud qui en est proche,
- une destination IPv6 interne : Bob.

Même si tous les routeurs IPv4 entre l'attaquant et le décapsuleur mettent en œuvre le filtrage d'entrée IPv4, et si tous les routeurs IPv6 entre le décapsuleur et Bob mettent en œuvre le filtrage d'entrée IPv6, les paquets falsifiés ci-dessus ne seront pas filtrés. Par suite, Bob va recevoir un paquet qui semble avoir été envoyé par Alice même si l'envoyeur est un nœud sans relation avec elle.

La solution à cela est de faire que le décapsuleur n'accepte que les paquets encapsulés provenant d'une adresse de source explicitement configurée (c'est-à-dire, l'autre extrémité du tunnel) comme spécifié au paragraphe 3.6. Bien que cela ne fournisse pas une complète protection dans le cas où le filtrage d'entrée n'a pas été déployé, cela fournit une augmentation significative de la sécurité. Le problème et le reste des menaces sont discutés plus en détails dans les considérations sur la sécurité.

5. Considérations sur la sécurité

Les considérations générales sur la sécurité de l'utilisation de IPv6 sont discutées dans un document séparé [RFC4942].

Une mise en œuvre de tunnelage doit savoir que bien qu'un tunnel soit une liaison (comme défini dans la [RFC2460]) le modèle de menaces pour un tunnel peut être assez différent de celui des autres liaisons, car le tunnel inclut potentiellement tout l'Internet.

Plusieurs mécanismes (par exemple, la découverte de voisin) dépendent du compte de bonds de 255 et/ou de ce que les adresses sont de liaison locale pour assurer qu'un paquet est originaire de la liaison, dans un environnement de demi confiance. Les tunnels sont plus vulnérables à une entorse à cette hypothèse que les liaisons physiques, car un attaquant peut de n'importe où dans l'Internet envoyer un paquet IPv6 dans IPv4 au décapsuleur du tunnel, causant l'injection d'un paquet IPv6 encapsulé à l'interface de tunnel configurée sauf si les vérifications de décapsulation sont capables d'éliminer les paquets injectés de cette façon.

Donc, le présent mémoire spécifie que les décapsuleurs suivent ces étapes (comme décrit au paragraphe 3.6) pour diminuer cette menace :
- l'adresse IPv4 de source du paquet DOIT être la même que celle configurée pour le point d'extrémité du tunnel ;
- indépendamment de tout filtrage d'entrée IPv4 que l'administrateur peut avoir configuré, la mise en œuvre PEUT effectuer un filtrage d'entrée IPv4 pour vérifier que les paquets IPv4 sont reçus d'une interface prévue (mais comme cela peut causer des problèmes, cela peut être désactivé par défaut) ;
- les paquets IPv6 avec plusieurs adresses de source IPv6 visiblement invalides reçus du tunnel DOIVENT être éliminés (voir les détails au paragraphe 3.6) ; et
- le filtrage d'entrée IPv6 devrait être effectué (exigeant normalement la configuration par l'opérateur) pour vérifier que les paquets IPv6 tunnelés sont reçus d'une interface prévue.

En particulier, la première vérification est vitale : pour éviter cette vérification, l'attaquant doit être capable de connaître la source du tunnel (ce qui va du difficile au prévisible) et être capable de se faire passer pour elle (plus facile).

Si le reste des menaces de la vérification de la source de tunnel est considéré comme significatif, un schéma de tunnelage avec authentification devrait être utilisé à la place, par exemple, IPsec [RFC2401] (préférable) ou l'encapsulation générique d'acheminement avec une clé secrète pré-configurée [RFC2890]. Comme les tunnels configurés sont établis plus ou moins manuellement, établir le matériel de chiffrement n'est probablement pas un problème. Cependant, établir un tunnel IPsec IPv6 dans IPv4 sûr est décrit dans un autre document [RFC4891].

Si le tunnelage est fait à l'intérieur d'un domaine administratif, le filtrage d'entrée approprié à la bordure du domaine peut aussi éliminer la menace de l'extérieur du domaine. Donc, les tunnels plus courts sont préférables aux plus longs, éventuellement s'étendant sur tout l'Internet.

De plus, une mise en œuvre DOIT traiter les interfaces à des liaisons différentes comme séparées, par exemple, pour s'assurer que les paquets de découverte de voisin qui arrivent sur une liaison n'affectent pas les autres liaisons. C'est particulièrement important pour les liaisons tunnels.

Lorsque on élimine des paquets parce qu'ils ne correspondent pas aux adresses de source IPv4 admises pour un tunnel, le nœud ne devrait pas "accuser réception" de l'existence d'un tunnel, cela pourrait autrement être utilisé pour sonder les

adresses acceptables de point d'extrémité de tunnel. Pour cette raison, la spécification dit que de tels paquets DOIVENT être éliminés, et qu'un message d'erreur ICMP NE DEVRAIT PAS être généré, sauf si la mise en œuvre envoie normalement des messages ICMP "Destination injoignable" pour les protocoles inconnus ; dans ce cas, le même code PEUT être envoyé. Comme cela devrait être évident, ne pas retourner le même code ICMP si une erreur est retournée pour d'autres protocoles peut donner l'indication que la pile IPv6 (ou le protocole de traitement de tunnelage 41 a été activé -- le comportement devrait être cohérent quand à la façon dont la mise en œuvre se comporte par ailleurs pour être transparente aux sondages.

6. Remerciements

Nous tenons à remercier les membres des groupes de travail IPv6, Next Generation Transition (ngtrans) et v6ops pour leurs nombreuses contributions et leur relecture extensive du présent document. Des remerciements particuliers sont dus (en ordre alphabétique) à Jim Bound, Ross Callon, Tim Chown, Alex Conta, Bob Hinden, Bill Manning, John Moy, Mohan Parthasarathy, Chirayu Patel, Pekka Savola, et Fred Templin pour leurs nombreuses suggestions utiles. Pekka Savola a aidé à l'édition des révisions finales de la spécification.

7. Références

7.1 Références normatives

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du protocole du programme Internet", STD 5, septembre 1981.

[RFC1191] J. Mogul et S. Deering, "Découverte de la MTU de chemin", novembre 1990.

[RFC1981] J. McCann, S. Deering, J. Mogul, "Découverte de la MTU de chemin pour IP version 6", août 1996. *(D.S. ; Remplacé par* [RFC8398], STD87)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2460] S. Deering et R. Hinden, "Spécification du protocole Internet, version 6 (IPv6)", décembre 1998. *(MàJ par* 5095, 6564 *; D.S ; Remplacée par* RFC8200, STD 86*)*

[RFC2463] A. Conta, S. Deering, "Protocole de message de contrôle Internet (ICMPv6) pour le protocole Internet version 6 (IPv6)", décembre 1998. (*Obsolète, voir* RFC4443) *(D.S.)*

7.2 Références pour information

[ASSIGNED] IANA, "Base de données en ligne des numéros alloués", http://www.iana.org/numbers.html

[KM97] Kent, C., and J. Mogul, "Fragmentation Considered Harmful". Dans Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology, août 1987.

[RFC1435] S. Knowles, "Avis de l'IESG d'une expérience avec découverte de la MTU de chemin", mars 1993. *(Info)*

[RFC1812] F. Baker, "Exigences pour les routeurs IP version 4", juin 1995. *((MàJ par les* RFC2644, RFC6633)

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2461] T. Narten, E. Nordmark, W. Simpson, "Découverte de voisins pour IP version 6 (IPv6)", décembre 1998. (*Obsolète, voir* RFC4861) *(D.S.)*

[RFC2462] S. Thomson, T. Narten, "Autoconfiguration d'adresse IPv6 sans état", décembre 1998. (*Obsolète, voir* RFC4862) *(D.S.)*

[RFC2827] P. Ferguson, D. Senie, "Filtrage d'entrée de réseau : Combattre les attaques de déni de service qui utilisent l'usurpation d'adresse de source IP", mai 2000. (*MàJ par* RFC3704) (BCP0038)

[RFC2890] G. Dommety, "Extensions de clé et de numéro de séquence à GRE", septembre 2000. *(P.S.)*

[RFC2923] K. Lahey, "Problèmes de TCP avec la découverte de MTU de chemin", septembre 2000. *(Information)*

[RFC2983] D. Black, "Services différenciés et tunnels", octobre 2000. *(Information)*

[RFC3056] B. Carpenter, K. Moore, "Connexion des domaines IPv6 via des nuages IPv4", février 2001. *(P.S.)*

[RFC3168] K. Ramakrishnan et autres, "Ajout de la notification explicite d'encombrement (ECN) à IP", septembre 2001. *(P.S. ; MàJ par* RFC8311)

[RFC3232] J. Reynolds, "Numéros alloués : la RFC 1700 est remplacée par une base de données en ligne", janvier 2002.

[RFC3484] R. Draves, "Choix d'adresse par défaut pour le protocole Internet version 6 (IPv6)", février 2003. (*Remplacée par la* RFC6724) *(P.S.)*

[RFC3493] R. Gilligan et autres, "Extensions d'interface de prise de base pour IPv6", février 2003. *(Information)*

[RFC3513] R. Hinden et S. Deering, "Architecture d'adressage du protocole Internet version 6 (IPv6)", avril 2003. (*Obs. voir* RFC4291)

[RFC3596] S. Thomson et autres, "Extensions au DNS pour la prise en charge de IPv6", octobre 2003. *(D.S.)*

[RFC3704] F. Baker, P. Savola, "Filtrage d'entrée pour réseaux à rattachement multiples", mars 2004. (BCP0084)

[RFC4087] D. Thaler, "MIB de tunnel IP", juin 2005. (*Remplace* RFC2667) *(P.S.)*

[RFC4472] A. Durand et autres, "Considérations et problèmes de fonctionnement du DNS IPv6", avril 2006. *(Info.)*

[RFC4891] R. Graveman et autres, "Utilisation d'IPsec pour sécuriser les tunnels IPv6 dans IPv4" mai 2007. *(Info.)*

[RFC4942] E. Davies et autres, "Considérations sur la sécurité pour la transition/co-existence avec IPv6", septembre 2007 *(Info.*

8. Changements par rapport à la RFC 2893

La motivation pour le plus gros de ces changements est de simplifier le document pour qu'il contienne seulement les mécanismes d'utilisation largement répandues.

La RFC 2893 contient un mécanisme appelé tunnelage automatique. Mais un mécanisme beaucoup plus général est spécifié dans la [RFC3056] qui donne à chaque nœud avec une adresse IPv4 (mondiale) un préfixe /48 IPv6, c'est-à-dire assez pour un site entier.

Les changements suivants ont été effectués depuis la RFC 2893:
- Suppression des références à A6 et conservation de AAAA.
- Suppression du tunnelage automatique et de l'utilisation des adresses compatibles IPv4.
- Suppression du tunnel configuré par défaut utilisant l'adresse de diffusion à la cantonade IPv4.
- Suppression de la section Sélection d'adresse de source car c'est maintenant couvert par un autre document ([RFC3484]).
- Suppression de la mention abrégée de 6sur4.
- Partage des références en normatives et non normatives et autres nettoyages des références.
- Abandon de "ou égal" dans "si (la MTU de chemin IPv4 - 20) est inférieure ou égale à 1280.
- Abandon de : "Cependant, IPv6 peut être utilisé dans certains environnements où l'interopérabilité avec IPv4 n'est pas exigée. Les nœuds IPv6 qui sont destinés à être utilisés dans de tels environnements n'ont pas besoin d'utiliser ou même mettre en œuvre ces mécanismes.
- Description séparée des cas de MTU statique et dynamique ; précision que le mécanisme de MTU dynamique est FACULTATIF mais si il est mis en œuvre, il devrait suivre les règles du paragraphe 3.2.2.
- Spécifié la MTU statique par défaut à une MTU de 1280 à 1480 octets, et que cela peut être configurable. Discussion des problèmes de l'utilisation de la MTU statique plus en détails.

- Spécifié les règles minimales du réassemblage IPv4 et de la MRU IPv6 pour améliorer l'interopérabilité et minimiser les trous noirs.
- Réaffirmation du langage "actuellement utilisé" sur le Type de service, et référence aux [RFC2983] et [RFC3168].
- Correction de la référence aux Numéros Alloués comme étant la version en ligne (avec un pointeur approprié sur la RFC "Les Numéros alloués sont obsolètes").
- Précision du texte sur le filtrage d'entrée qui, par exemple, s'applique au paquet livré aux protocoles de transport sur le décapsuleur aussi bein qu'aux paquets transmis par le décapsuleur, et comment les vérifications du décapsuleur aident quand le filtrage d'entrée IPv4 et IPv6 est en place.
- Suppression du tunnelage unidirectionnel ; on suppose que tous les tunnels sont bidirectionnels, entre les adresses de point d'extrémité (pas les nœuds).
- Suppression des lignes directrices pour l'annonce des adresses dans le DNS comme légèrement hors sujet, se référant à un autre document pour les détails.
- Suppression de l'exigence DEVRAIT que les adresses locales de liaisons devraient être formées sur la base des adresses IPv4.
- Ajout d'un DEVRAIT pour mettre en œuvre un bouton pour être capable de régler l'adresse de source du tunnel, et ajout de la discussion de pourquoi c'est utile.
- Ajout d'une formulation plus forte pour les vérifications d'adresse de source : les adresses de source IPv4 et IPv6 DOIVENT toutes deux être vérifiées, et le filtrage d'entrée de style RPF est facultatif.
- Réécriture des considérations sur la sécurité pour être plus précis sur les menaces pour le tunnelage.
- Ajout d'une note sur le fait d'envisager un TTL=255 lors de l'encapsulation.
- Ajout de plus de discussion au paragraphe 3.2 sur pourquoi utiliser une MTU IPv6 "infinie" conduit à de probables problèmes d'interopérabilité.
- Ajout de l'exigence explicite que si les deux méthodes de détermination de la MTU sont utilisées, le choix d'une devrait être possible tunnel par tunnel.
- Précision que le traitement d'erreur ICMPv4 n'est applicable qu'à la détermination de MTU dynamique.
- Suppression/précision du filtrage d'enregistrement DNS ; une API est un DEVRAIT et si il n'en existe pas, NE DOIT PAS filtrer qui que ce soit. L'ordre de présentation est hors sujet, mais se référer à la RFC3484.
- Ajout d'une note que l'adresse de destination IPv4 pourrait aussi être une adresse de diffusion groupée.
- Il est RECOMMANDÉ de fournir une bascule pour effectuer un strict filtrage d'entrée sur une interface.
- Généralisation du texte sur les données dans les messages ICMPv4.
- Effectué un nettoyage général de diverses imprécisions rédactionnelles.

Adresse des auteurs

Erik Nordmark
Sun Microsystems
17 Network Circle
Menlo Park, CA 94025
USA
téléphone : +1 650 786 2921
mél : erik.nordmark@sun.com

Robert E. Gilligan
Intransa, Inc.
2870 Zanker Rd., Suite 100
San Jose, CA 95134 USA
téléphone : +1 408 678 8600
mél : bob.gilligan@acm.org

Déclaration complète de droits de reproduction

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 le 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 la Internet Society.