Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 8200
STD : 86
RFC rendue obsolète : 2460
Catégorie : Norme
ISSN: 2070-1721

S. Deering, retraité
R. Hinden, Check Point Software

juillet 2017

Traduction Claude Brière de L'Isle

Spécification de la version 6 du protocole Internet (IPv6)

Résumé
Le présent document spécifie la version 6 du protocole Internet (IPv6). Il rend obsolète la RFC 2460.

Statut du présent mémoire
Ce document est sur la voie de la normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF, *Internet Engineering Task Force*). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG, *Internet Engineering Steering Group*). D'autres informations sur les normes de l'Internet sont disponibles à la Section 2 de la RFC 7841.

Des informations sur le statut actuel de ce document, les errata éventuels, et comment y contribuer peuvent être obtenues à http://www.rfc-editor.org/info/rfc8200.

Table des matières

1. Introduction

IP version 6 (IPv6) est une nouvelle version du protocole Internet (IP, *Internet Protocol*) conçue comme successeur de IP version 4 (IPv4) [RFC0791]. Les changements de IPv4 à IPv6 entrent principalement dans les catégories suivantes :

- Extension des capacités d'adressage : IPv6 augmente la taille de l'adresse IP de 32 bits à 128 bits, pour prendre en charge plus de niveaux de hiérarchie d'adressage, un plus grand nombre de nœuds adressables, et une autoconfiguration plus simple des adresses. L'adaptabilité de l'acheminement de diffusion groupée est améliorée par l'ajout d'un champ "Portée" aux adresses de diffusion groupée. Et un nouveau type d'adresse appelé une "adresse d'envoi à la cantonade" est défini ; il est utilisé pour envoyer un paquet à tous les nœuds d'un groupe.

- Simplification du format d'en-tête : certains champs d'en-tête IPv4 ont été abandonnés ou rendus facultatifs pour réduire le coût de traitement du paquet ordinaire et pour limiter le coût en bande passante de l'en-tête IPv6.

- Amélioration de la prise en charge des extensions et des options : les changements de la façon dont sont codées les options d'en-tête IP permettent une transmission plus efficace, des limites moins contraignantes sur la longueur des options, et une plus grande souplesse pour introduire de nouvelles options à l'avenir.

- Capacité d'étiquetage de flux : une nouvelle capacité est ajoutée pour permettre l'étiquetage des séquences de paquets dont l'envoyeur demande au réseau de les traiter comme un seul flux.

- Capacités d'authentification et de confidentialité : des extensions pour prendre en charge l'authentification, la protection de l'intégrité des données, et (facultativement) de la confidentialité des données sont spécifiées pour IPv6.

Le présent document spécifie l'en-tête IPv6 de base et les en-têtes d'extension et options IPv6 initialement définis. Il discute aussi des questions de taille de paquet, de la sémantique des étiquettes de flux et des classes de trafic, et des effets de IPv6 sur les protocoles de couche supérieure. Le format et la sémantique des adresses IPv6 sont spécifiés séparément dans la [RFC4291]. La version IPv6 de ICMP, dont l'inclusion est exigée de toutes les mises en œuvre de IPv6, est spécifiée dans la [RFC4443].

L'ordre de transmission des données pour IPv6 est le même que pour IPv4 comme défini à l'Appendice B de la [RFC0791].

Note : comme le présent document rend obsolète la [RFC2460], tout document référencé dans ce document qui inclut des pointeurs sur la RFC 2460 devrait être interprété comme faisant référence au présent document.

2. Terminologie

nœud : appareil qui met en œuvre IPv6.

routeur : nœud qui transmet des paquets IPv6 non explicitement adressés à lui-même. (Voir la note ci-dessous.)

hôte : tout nœud qui n'est pas un routeur. (Voir la note ci-dessous.)

couche supérieure : couche de protocole immédiatement au dessus de IPv6. Des exemples sont des protocoles de transport comme TCP et UDP, des protocoles de contrôle comme ICMP, des protocoles d'acheminement comme OSPF, et des protocoles de couche internet ou en dessous qui sont "tunnelés" sur (c'est-à-dire, encapsulés dans) IPv6 comme l'échange de paquets inter réseaux (IPX, *Internetwork Packet Exchange*), AppleTalk, ou IPv6 lui-même.

liaison : facilité ou support de communication sur lequel les nœuds peuvent communiquer à la couche de liaison, c'est-à-dire, à la couche immédiatement en-dessous de IPv6. Des exemples sont les Ethernets (simples ou pontés) les liaisons PPP, X.25, le relais de trame, ou les réseaux ATM, et les "tunnels" de couche internet ou supérieure, comme les tunnels sur IPv4 ou IPv6 lui-même.

voisins : nœuds rattachés à la même liaison.

interface : rattachement d'un nœud à une liaison.

adresse : identifiant de couche IPv6 pour une interface ou un ensemble d'interfaces.

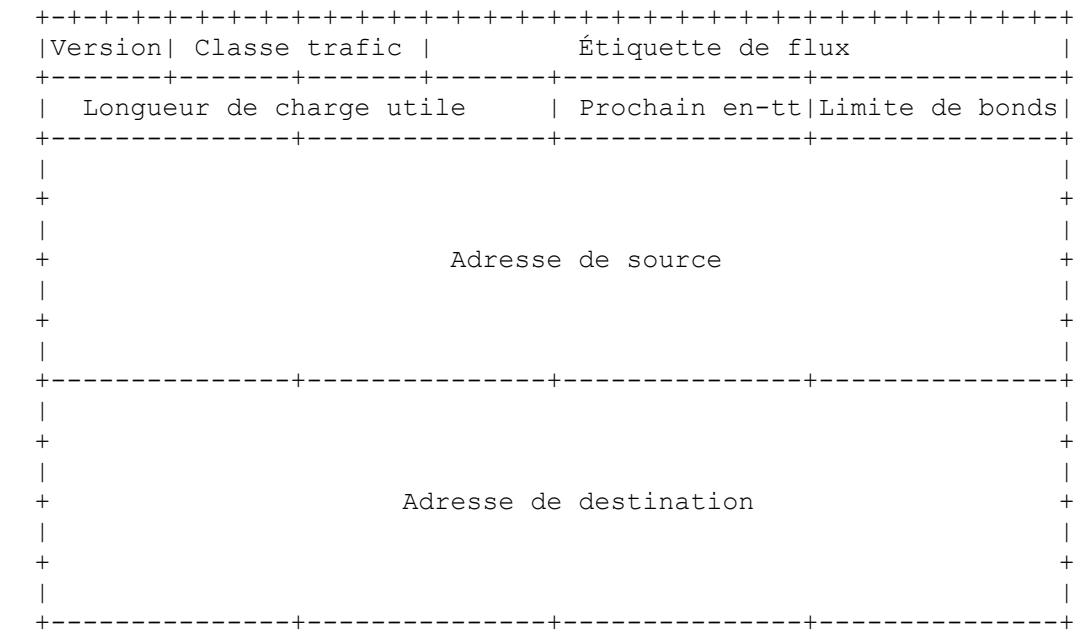
paquet : un en-tête IPv6 plus une charge utile.

MTU de liaison : unité maximum de transmission, c'est-à-dire, taille maximum de paquet en octets, qui peut être convoyée sur une liaison.

MTU de chemin : MTU minimum de liaison de toutes les liaisons dans un chemin entre un nœud de source et un nœud de destination.

Note : il est possible qu'un appareil avec plusieurs interfaces soit configuré à transmettre des paquets qui ne lui sont pas destinés qui arrivent d'un certain ensemble de ses interfaces (moins que toutes) et à éliminer les paquets qui ne lui sont pas destinés qui arrivent de ses autres interfaces. Un tel appareil doit respecter les exigences de protocole des routeurs lorsque il reçoit des paquets de voisins avec lesquels il interagit, de ces premières interfaces (en émission). Il doit respecter les exigences de protocole pour les hôtes quand il reçoit des paquets de, et interagit avec ses voisins sur, les dernières interfaces (non en émission).

3. Format d'en-tête IPv6



Version : numéro de version de 4 bits de protocole Internet = 6.
Classe trafic : champ de 8 bits de classe de trafic. Voir la Section 7.
Étiquette de flux : étiquette de flux de 20 bits. Voir la Section 6.
Longueur de charge utile : entier non signé de 16 bits. Longueur de la charge utile IPv6, c'est-à-dire, le reste du paquet qui

suit cet en-tête IPv6, en octets. (Noter que tout en-tête d'extension (voir la Section 4) présent est considéré faire partie de la charge utile, c'est-à-dire, est inclus dans le compte de longueur.)

Prochain en-tête : sélecteur de 8 bits. Identifie le type d'en-tête qui suit immédiatement l'en-tête IPv6. Utilise les mêmes valeurs que le champ Protocole IPv4 [IANA-PN].

Limite de bond : entier non signé de 8 bits. Décrémenté de 1 à chaque nœud qui transmet le paquet. À la transmission, le paquet est éliminé si la limite de bond était zéro à la réception ou si il est décrémenté à zéro. Un nœud qui est la destination d'un paquet ne devrait pas éliminer un paquet dont la limite de bond est égale à zéro ; il devrait traiter le paquet normalement.

Adresse de source : adresse de 128 bits du générateur du paquet. Voir la [RFC4291].

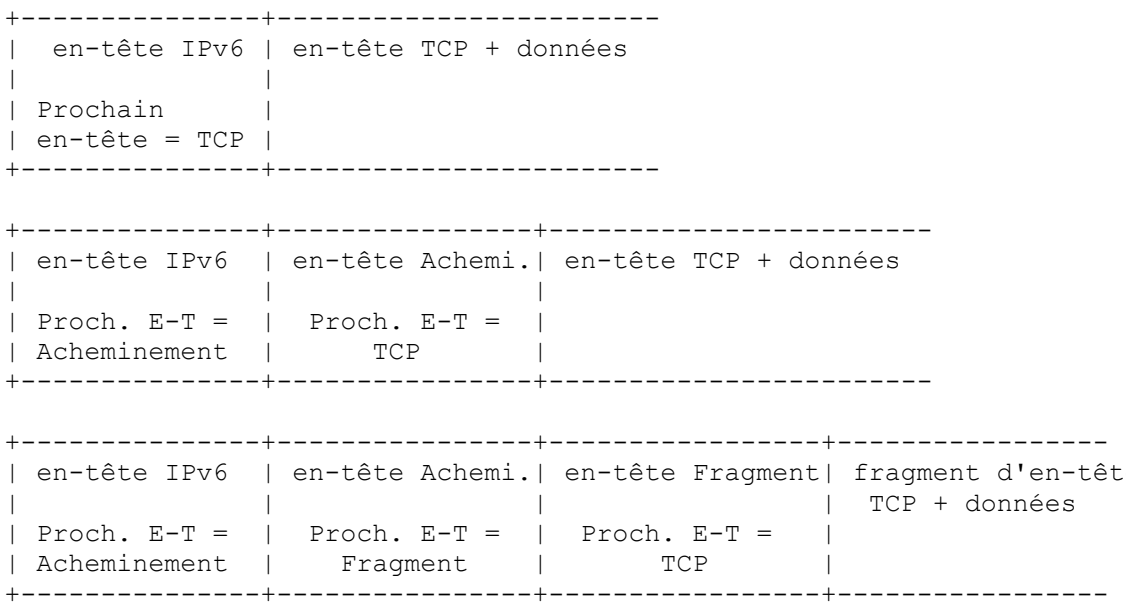
Adresse de destination : adresse de 128 bits du receveur prévu du paquet (éventuellement pas le receveur ultime, si un en-tête d'acheminement est présent). Voir la [RFC4291] et le paragraphe 4.4.

4. En-têtes d'extension IPv6

Dans IPv6, des informations facultatives de couche internet sont codées dans des en-têtes séparés qui peuvent être placés entre l'en-tête IPv6 et l'en-tête de couche supérieure dans un paquet. Il y a un petit nombre de ces en-têtes d'extension, chacun identifié par une valeur distincte de Prochain en-tête.

Les en-têtes d'extensions sont numérotés à partir des numéros de protocole IP de l'IANA [IANA-PN], les mêmes valeurs sont utilisées pour IPv4 et IPv6. Quand on traite une séquence de valeurs de prochain en-tête dans un paquet, la première qui n'est pas un en-tête d'extension [IANA-EH] indique que le prochain élément dans le paquet est l'en-tête de couche supérieure correspondant. Une valeur spéciale "Pas de prochain en-tête" est utilisée si il n'y a pas d'en-tête de couche supérieure.

Comme illustré dans ces exemples, un paquet IPv6 peut porter zéro, un, ou plusieurs en-têtes d'extension, identifiés chacun par le champ Prochain en-tête de l'en-tête précédent :



Les en-têtes d'extension (sauf pour les en-têtes d'options bond par bond) ne sont pas traités, insérés, ou supprimés par un nœud le long du chemin de livraison du paquet, jusqu'à ce que le paquet atteigne le nœud (ou chacun des ensembles de nœuds, dans le cas de diffusion groupée) identifié dans le champ Adresse de destination de l'en-tête IPv6.

L'en-tête d'options bond par bond n'est pas inséré ni supprimé, mais peut être examiné ou traité par tout nœud le long du chemin de livraison d'un paquet, jusqu'à ce que le paquet atteigne le nœud (ou chacun des ensembles de nœuds, dans le cas de diffusion groupée) identifié dans le champ Adresse de destination de l'en-tête IPv6. L'en-tête d'options bond par bond, lorsque il est présent, doit suivre immédiatement l'en-tête IPv6. Sa présence est indiquée par la valeur zéro dans le champ Prochain en-tête de l'en-tête IPv6.

Note : Alors que la [RFC2460] exigeait que tous les nœuds examinent et traitent l'en-tête Options bond par bond, il est

maintenant attendu que les nœuds le long du chemin de livraison d'un paquet examinent seulement et traitent l'en-tête Options bond par bond si ils sont explicitement configurés à le faire. Au nœud de destination, le démultiplexage normal du champ Prochain en-tête de l'en-tête IPv6 invoque le module pour traiter le premier en-tête d'extension, ou l'en-tête de couche supérieure si aucun en-tête d'extension n'est présent. Le contenu et la sémantique de chaque en-tête d'extension détermine si il faut traiter ou non le prochain en-tête. Donc, les en-têtes d'extension doivent être traités strictement dans l'ordre où ils apparaissent dans le paquet ; un receveur ne doit pas, par exemple, examiner un paquet à la recherche d'une sorte particulière d'en-tête d'extension et traiter cet en-tête avant de traiter tous les précédents.

Si, par suite du traitement d'un en-tête, le nœud de destination est obligé de traiter le prochain en-tête mais si la valeur de prochain en-tête dans l'en-tête courant n'est pas reconnue par le nœud, il devrait éliminer le paquet et envoyer un message ICMP "Problème de paramètre" à la source du paquet, avec une valeur de code ICMP de 1 "Type de prochain en-tête non reconnu" et le champ Pointeur ICMP contenant le décalage de la valeur non reconnue au sein du paquet d'origine. La même action devrait être prise si un nœud rencontre une valeur de prochain en-tête de zéro dans un en-tête autre qu'un en-tête IPv6.

Chaque en-tête d'extension est un entier multiple de 8 octets, afin de conserver un alignement sur 8 octets pour les en-têtes suivants. Les champs de plusieurs octets au sein de chaque en-tête d'extension sont alignés sur leurs frontières naturelles, c'est-à-dire, les champs de longueur n octets sont placés à un multiple entier de n octets du début de l'en-tête, pour n = 1, 2, 4, ou 8.

Une pleine mise en œuvre de IPv6 inclut la mise en œuvre des en-têtes d'extension suivants :
Options bond par bond
Fragment
Options de destination
Acheminement
Authentification
Encapsulation de charge utile de sécurité

Les quatre premiers sont spécifié dans le présent document ; les deux derniers sont spécifiés dans les [RFC4302] et [RFC4303], respectivement. La liste actuelle des en-têtes d'extension IPv6 se trouve dans [IANA-EH].

4.1 Ordre des en-têtes d'extension

Quand plus d'un en-tête d'extension est utilisé dans le même paquet, il est recommandé que ces en-têtes apparaissent dans l'ordre suivant :
en-tête IPv6
en-tête Options bond par bond
en-tête Options de destination (note 1)
en-tête Acheminement
en-tête Fragment
en-tête Authentification (note 2)
en-tête Encapsulation de charge utile de sécurité (note 2)
en-tête Options de destination (note 3)
en-tête de couche supérieure

note 1 : pour les options à traiter par la première destination qui apparaît dans le champ Adresse de destination IPv6 plus les destinations suivantes mentionnées dans l'en-tête Acheminement.

note 2 : des recommandations supplémentaires concernant l'ordre relatif des en-têtes Authentification et Encapsulation de charge utile de sécurité sont données dans la [RFC4303].

note 3 : pour les options à ne traiter que la par la destination finale du paquet.

Chaque en-tête d'extension devrait se produire au plus une fois, sauf pour l'en-tête Options de destination, qui devrait survenir au plus deux fois (une fois avant un en-tête Acheminement et une fois avant l'en-tête de couche supérieure).

Si l'en-tête de couche supérieure est un autre en-tête IPv6 (dans le cas de IPv6 tunnelé sur, ou encapsulé dans IPv6) il peut être suivi par ses propres en-têtes d'extension, qui sont séparément soumis aux mêmes recommandations d'ordre.

Si et quand d'autres en-têtes d'extension sont définis, leurs contraintes d'ordre par rapport à la liste des en-têtes ci-dessus doivent être spécifiées.

Les nœuds IPv6 doivent accepter et tenter de traiter les en-têtes d'extension dans tout ordre et se produisant tout nombre de fois dans le même paquet, excepté pour l'en-tête Options bond par bond, qui est contraint d'apparaître seulement immédiatement après un en-tête IPv6. Néanmoins, il est fortement recommandé que les sources des paquets IPv6 adhèrent à l'ordre recommandé ci-dessus tant que des spécifications ultérieures n'auront pas révisé cette recommandation.

4.2 Options

Deux des en-têtes d'extension actuellement définis spécifiés dans ce document -- l'en-tête Options bond par bond et l'en-tête Options de destination -- portent un nombre variable "d'options" qui sont codées en type-longueur-valeur (TLV) dans le format suivant :

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
| Type d'option | Longueur      | Données d'option
+---------------+---------------+- - - - - - - - -
```

Type d'option : identifiant de 8 bits du type d'option.

Longueur : entier non signé de 8 bits. Longueur du champ Données d'option de cette option, en octets.

Données d'option : champ de longueur variable. Données spécifiques du type d'option.

La séquence des options au sein d'un en-tête doit être traitée strictement dans l'ordre dans lequel elles apparaissent dans l'en-tête ; un receveur ne doit pas, par exemple, examiner l'en-tête à la recherche d'une sorte d'option particulière et traiter cette option avant de traiter toutes les précédentes.

Les identifiants de type d'option sont codés en interne de telle façon que leurs 2 bits de poids fort spécifient l'action qui doit être effectuée si le nœud IPv6 ne reconnaît pas le type d'option :
00 - sauter cette option et continuer le traitement de l'en-tête.
01 - éliminer le paquet.
10 - éliminer le paquet et, sans considérer si l'adresse de destination du paquet est ou non une adresse de diffusion groupée, envoyer un message ICMP Problème de paramètre, code 2, à l'adresse de source du paquet, pointant le type d'option non reconnu.
11 - éliminer le paquet et, seulement si l'adresse de destination du paquet n'était pas une adresse de diffusion groupée, envoyer un message ICMP Problème de paramètre, code 2, à l'adresse de source du paquet, pointant sur le type d'option non reconnu.

Le troisième bit de plus fort poids du type d'option spécifie si les données d'option de cette option peuvent ou non changer en route pour la destination finale du paquet. Quand un en-tête Authentification est présent dans le paquet, pour toute option dont les données peuvent changer en route, son champ Données d'option entier doit être traité comme des octets tout à zéro lors du calcul ou de la vérification de la valeur qui authentifie le paquet.

0 - Les données d'option ne changent pas en route
1 - Les données d'option peuvent changer en route

Les trois bits de poids fort décrits ci-dessus sont à traiter au titre du type d'option, et pas indépendamment du type d'option. c'est-à-dire qu'une option particulière est identifiée par un Type d'option de 8 bits complet, et non seulement juste les 5 bits de moindre poids d'un type d'option.

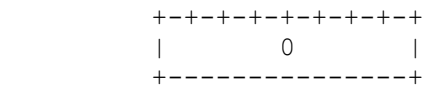
Le même espace de numérotation de type d'option est utilisé pour l'en-tête Options bond par bond et l'en-tête Options de destination. Cependant, la spécification d'une option particulière peut restreindre son utilisation à seulement un de ces deux en-têtes.

Des options individuelles peuvent avoir des exigences d'alignement spécifiques, pour assurer que des valeurs de plusieurs octets au sein des champs de données d'option tombent sur des limites naturelles. L'exigence d'alignement d'une option est spécifiée en utilisant la notation xn+y, ce qui signifie que le type d'option doit apparaître à un multiple entier de x octets à partir du début de l'en-tête, plus y octets. Par exemple :
2n signifie tout décalage de 2 octets à partir du début de l'en-tête.

8n+2 signifie tout décalage de 8 octets à partir du début de l'en-tête, plus 2 octets.

Il y a deux options de bourrage qui sont utilisées quand il est nécessaire d'aligner les options suivantes et pour bourrer l'en-tête contenant sur une longueur multiple de 8 octets. Ces options de bourrage doivent être reconnues par toutes les mises en œuvre de IPv6 :

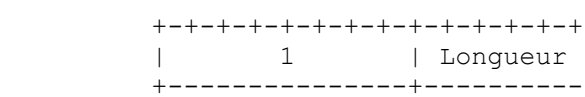
Option Pad1 (exigence d'alignement : aucune)



Note : le format de l'option Pad1 est un cas particulier -- il n'a pas de champs Longueur et Valeur.

L'option Pad1 est utilisée pour insérer 1 octet de bourrage dans la zone Options d'un en-tête. Si plus d'un octet de bourrage est requis, l'option PadN, décrite ensuite, devrait être utilisée, plutôt que plusieurs options Pad1.

Option PadN (exigence d'alignement : aucune)

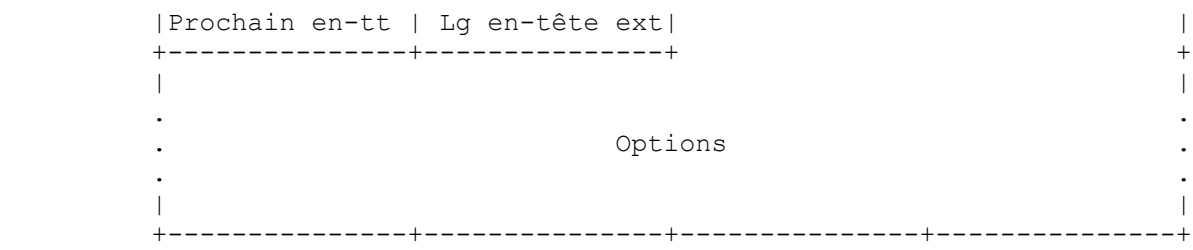


L'option PadN est utilisée pour insérer deux octets ou plus de bourrage dans la zone Options d'un en-tête. Pour N octets de bourrage, le champ Longueur contient la valeur N-2, et les données d'option consistent en N-2 octets de valeur zéro.

L'Appendice A contient des lignes directrices de formatage pour la conception de nouvelles options.

4.3 En-tête Options bond par bond

L'en-tête Options bond par bond est utilisé pour porter des informations facultatives qui peuvent être examinées et traitées par chaque nœud le long du chemin de livraison d'un paquet. L'en-tête Options bond par bond est identifié par une valeur de prochain en-tête de 0 dans l'en-tête IPv6 et a le format suivant :



Prochain en-tête : sélecteur de 8 bits. Identifie le type de l'en-tête qui suit immédiatement l'en-tête Options bond par bond. Il utilise les mêmes valeurs que le champ Protocole IPv4 [IANA-PN].

Longueur en-tête externe : entier non signé de 8 bits. Longueur de l'en-tête Options bond par bond en unités de 8 octets, non inclus les huit premiers octets.

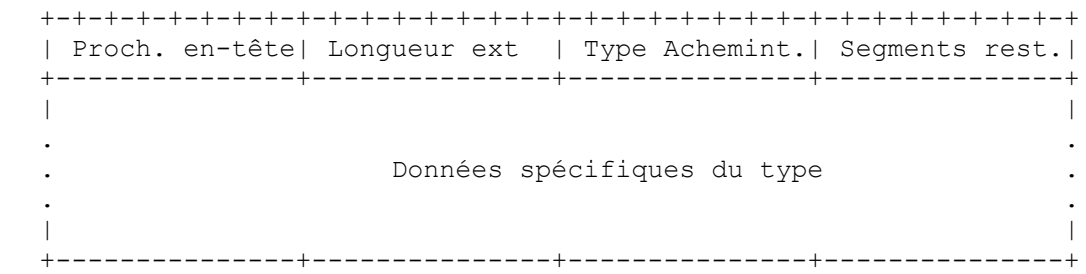
Options : champ de longueur variable, d'une longueur telle que l'en-tête Options bond par bond complet soit un multiple entier de 8 octets. Contient une ou plusieurs options codées en TLV, comme décrit au paragraphe 4.2.

Les seules options bond par bond définies dans ce document sont les options Pad1 et PadN spécifiées au paragraphe 4.2.

4.4. En-tête Acheminement

L'en-tête Acheminement est utilisé par une source IPv6 pour mentionner un ou plusieurs nœuds intermédiaires à "visiter" sur le chemin de la destination d'un paquet. Cette fonction est très similaire à l'option IPv4 de Source lâche et Enregistrement de chemin. L'en-tête Acheminement est identifié par une valeur de prochain en-tête de 43 dans l'en-tête

immédiatement précédant et a le format suivant :



Prochain en-tête : sélecteur de 8 bits. Identifie le type de l'en-tête qui suit immédiatement l'en-tête Acheminement. Utilise les mêmes valeurs que le champ Protocole IPv4 [IANA-PN].

Longueur ext : entier non signé de 8 bits. Longueur de l'en-tête Acheminement en unités de 8 octets, non inclus les 8 premiers octets.

Type d'acheminement : identifiant de 8 bits d'une variante particulière d'en-tête Acheminement.

Segments restants : entier non signé de 8 bits. Nombre de segments de chemin restants, c'est-à-dire, nombre de nœuds intermédiaires explicitement mentionnés à visiter avant d'atteindre la destination finale.

Données spécifiques du type : champ de longueur variable, de format déterminé par le type d'acheminement, et d'une longueur telle que l'en-tête Acheminement complet soit un multiple entier de 8 octets.

Si lors du traitement d'un paquet reçu, un nœud rencontre un en-tête Acheminement qui a une valeur de type d'acheminement non reconnue, le comportement exigé du nœud dépend de la valeur du champ Segments restants, comme suit :

Si Segments restants est zéro, le nœud doit ignorer l'en-tête Acheminement et passer au traitement du prochain en-tête dans le paquet, dont le type est identifié par le champ Prochain en-tête dans l'en-tête Acheminement.

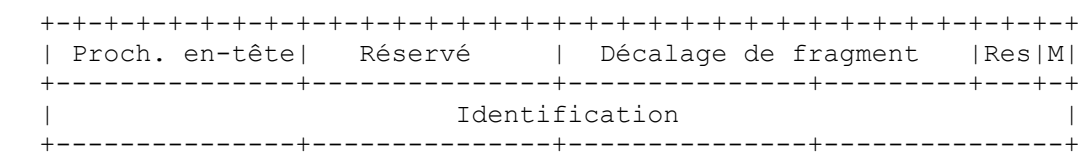
Si Segments restants n'est pas zéro, le nœud doit éliminer le paquet et envoyer un message ICMP Problème de paramètre, code 0, à l'adresse de source du paquet, pointant sur le type d'acheminement non reconnu.

Si, après le traitement de l'en-tête Acheminement d'un paquet reçu, un nœud intermédiaire détermine que le paquet est à transmettre sur une liaison dont la MTU de liaison est inférieure à la taille du paquet, le nœud doit éliminer le paquet et envoyer un message ICMP Paquet trop gros à l'adresse de source du paquet.

Les en-têtes Acheminement IPv6 actuellement définis et leur état se trouvent dans le registre [IANA-RH]. Des lignes directrices pour l'allocation des en-têtes Acheminement IPv6 se trouvent dans la [RFC5871].

4.5 En-tête Fragment

L'en-tête Fragment est utilisé par une source IPv6 pour envoyer un paquet plus gros que ce qui tiendrait dans la MTU de chemin vers sa destination. (Note : à la différence de IPv4, la fragmentation dans IPv6 n'est effectuée que par les nœuds de source, et non par les routeurs le long du chemin de livraison d'un paquet -- voir la Section 5.) L'en-tête Fragment est identifié par une valeur de prochain en-tête de 44 dans l'en-tête précédant immédiatement et a le format suivant :



Proch. en-tête : sélecteur de 8 bits. Identifie le type d'en-tête initial de la partie fragmentable du paquet d'origine (définie ci-dessous). Utilise les mêmes valeurs que le champ Protocole IPv4 [IANA-PN].

Réservé : champ de 8 bits réservé. Initialisé à zéro à l'émission; ignoré à réception.

Décalage de fragment : entier non signé de 13 bits. Le décalage, en unités de 8 octets, des données qui suivent cet en-tête, par rapport au début de la partie fragmentable du paquet d'origine.

Res : champ réservé de 2 bits. Initialisé à zéro à l'émission; ignoré à réception.

fanion M : 1 = plus de fragments ; 0 = dernier fragment.

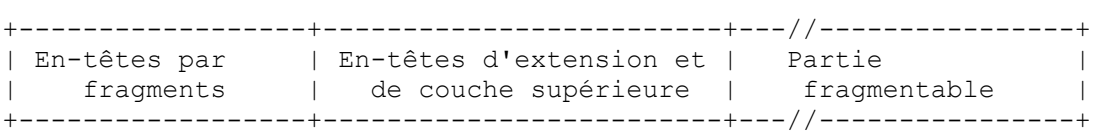
Identification : 32 bits. Voir la description ci-dessous.

Pour envoyer un paquet trop gros pour tenir dans la MTU de chemin vers la destination, un nœud source peut diviser le paquet en fragments et envoyer chaque fragment dans un paquet séparé, pour être ré-assemblés chez le receveur.

Pour chaque paquet à fragmenter, le nœud source génère une valeur d'identification. L'identification doit être différente de celle de tout autre paquet fragmenté envoyé récemment* avec les mêmes adresses de source et destination. Si un en-tête Acheminement est présent, l'adresse de destination concernée est celle de la destination finale.
* "récemment" signifie "dans la durée de vie maximum probable d'un paquet", incluant le temps de transit de la source à la destination et le temps passé à attendre le ré-assemblage avec les autres fragments du même paquet. Cependant, il n'est pas exigé qu'un nœud source connaisse la durée de vie maximum du paquet. Il est plutôt supposé que l'exigence peut être satisfaite par la mise en œuvre d'un algorithme qui résulte en une faible fréquence de réutilisation des identifications. Des exemples des algorithmes qui peuvent satisfaire cette exigence sont décrits dans la [RFC7739].

Le paquet initial, grand, non fragmenté est appelé le "paquet d'origine", et il est considéré comporter trois parties, comme le montre l'illustration suivante :

paquet d'origine :



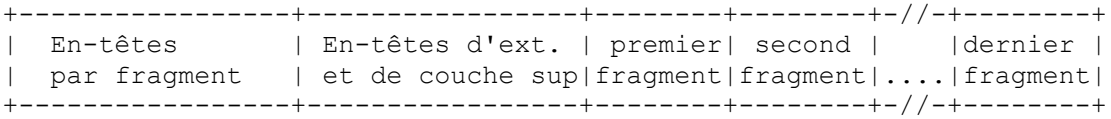
Les en-têtes par fragment doivent consister en l'en-tête IPv6 plus tous les en-têtes d'extension qui doivent être traités par les nœuds en route pour la destination, c'est-à-dire, tous les en-têtes jusqu'à et incluant l'en-tête Acheminement, si il est présent, autrement l'en-tête Options bond par bond, si il est présent, autrement aucun en-tête d'extension.

Les en-têtes d'extension sont tous les autres en-têtes d'extension qui ne sont pas inclus dans les en-têtes par fragment qui font partie du paquet. À cette fin, la charge utile de sécurité encapsulante (ESP) n'est pas considérée comme un en-tête d'extension. L'en-tête de couche supérieure est le premier en-tête de couche supérieure qui n'est pas un en-tête d'extension IPv6. Des exemples d'en-têtes de couche supérieure incluent TCP, UDP, IPv4, IPv6, ICMPv6, et ESP comme on vient de l'indiquer.

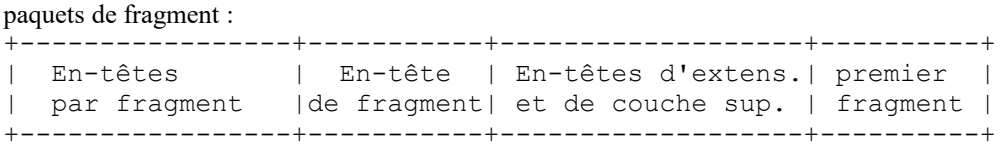
La partie fragmentable consiste en le reste du paquet après l'en-tête de couche supérieure ou après tout en-tête (c'est-à-dire, en-tête IPv6 initial ou en-tête d'extension) qui contient une valeur de prochain en-tête de Pas de prochain en-tête. La partie fragmentable du paquet d'origine est divisée en fragments. Les longueurs des fragments doivent être choisies de telle sorte que les paquets de fragment résultants tiennent dans la MTU du chemin de la destination du paquet. Chaque fragment complet, excepté éventuellement le dernier ("le plus à droite") est un multiple entier de 8 octets.

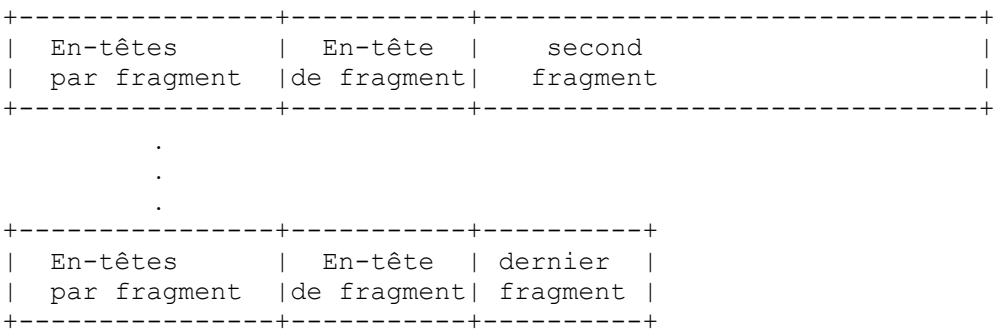
Les fragments sont transmis dans des "paquets de fragment" séparés, comme illustré ci-dessous :

paquet d'origine :



paquets de fragment :





Le premier paquet de fragment est composé de :

(1) Les en-têtes par fragment du paquet d'origine, avec la longueur de charge utile de l'en-tête IPv6 d'origine changée pour contenir la longueur de ce paquet de fragment seulement (en excluant la longueur de l'en-tête IPv6 lui-même) et le champ Prochain en-tête du dernier en-tête des en-têtes par fragment changé en 44.

(2) Un en-tête Fragment contenant :
- La valeur du prochain en-tête qui identifie le premier en-tête après les en-têtes par fragment du paquet d'origine.
- Un décalage de fragment contenant le décalage du fragment, en unités de 8 octets, par rapport au début de la partie fragmentable du paquet d'origine. Le décalage de fragment du premier fragment ("le plus à gauche") est 0.
- Une valeur de fanion M de 1 car c'est le premier fragment.
- La valeur d'identification générée pour le paquet d'origine.

(3) Les en-têtes d'extension, si il y en a, et l'en-tête de couche supérieure. Ces en-têtes doivent être dans le premier fragment. Note : Ceci restreint la taille des en-têtes jusqu'à l'en-tête de couche supérieure à la MTU du chemin de la destination du paquet.

(4) Le premier fragment.

Les paquets de fragment suivants sont composés de :

(1) Les en-têtes par fragment du paquet d'origine, avec la longueur de charge utile de l'en-tête IPv6 original changée pour contenir la longueur de ce seul paquet de fragment (en excluant la longueur de l'en-tête IPv6 lui-même) et le champ Prochain en-tête du dernier en-tête des en-têtes par fragment changé en 44.

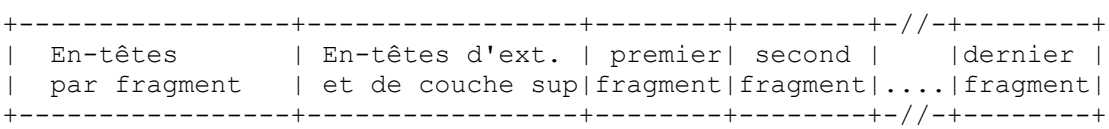
(2) Un en-tête Fragment contenant :
- La valeur du prochain en-tête qui identifie le premier en-tête après les en-têtes par fragment du paquet d'origine.
- Un décalage de fragment contenant le décalage du fragment, en unités de 8 octets, par rapport au début de la partie fragmentable du paquet d'origine.
- Une valeur de fanion M de 0 si le fragment est le dernier ("le plus à droite") sinon, une valeur de fanion M de 1.
- La valeur d'identification générée pour le paquet d'origine.

(3) Le fragment lui-même.

Les fragments ne doivent pas être créés en se chevauchant avec d'autres fragments créés à partir du paquet d'origine.

À la destination, les paquets de fragment sont ré-assemblés en leur forme originale, non fragmentée, comme illustré ici :

Paquet ré-assemblé original :



Les règles suivantes gouvernent le ré-assemblage :

Un paquet original n'est ré-assemblé qu'à partir des paquets de fragments qui ont la même adresse de source, de destination

et la même identification de fragment.

Les en-têtes par fragment du paquet ré-assemblé consistent en tous les en-têtes jusqu'à, non inclus, l'en-tête Fragment du premier paquet de fragment (c'est-à-dire, le paquet dont le décalage de fragment est zéro) avec les deux changements suivants :

Le champ Prochain en-tête du dernier en-tête des en-têtes par fragment est obtenu du champ Prochain en-tête de l'en-tête Fragment du premier fragment'.

La longueur de charge utile du paquet ré-assemblé est calculée à partir de la longueur des en-têtes par fragment et du décalage du dernier fragment. Par exemple, une formule pour calculer la longueur de charge utile du paquet d'origine ré-assemblé est :

PL.orig = PL.premier - FL. premier - 8 + (8 * FO.dernier) + FL.dernier

où
PL.orig = champ Longueur de charge utile du paquet ré-assemblé.
PL.premier = champ Longueur de charge utile du premier paquet de fragment.
FL.premier = longueur du fragment qui suit l'en-tête Fragment du premier paquet de fragment.
FO.dernier = champ Décalage de fragment de l'en-tête Fragment du dernier paquet de fragment.
FL.dernier = longueur du fragment qui suit l'en-tête Fragment du dernier paquet de fragment.

La partie fragmentable du paquet ré-assemblé est construite à partir des fragments qui suivent les en-têtes Fragment dans chacun des paquets de fragment. La longueur de chaque fragment est calculée en soustrayant de la longueur de charge utile du paquet la longueur des en-têtes entre l'en-tête IPv6 et le fragment lui-même ; sa position relative dans la partie fragmentable est calculée à partir de sa valeur de décalage de fragment.

L'en-tête Fragment n'est pas présent dans le paquet final ré-assemblé.

Si le fragment est un datagramme complet (c'est-à-dire, le champ Décalage de fragment et le fanion M sont tous deux à zéro) alors il n'y a pas besoin d'autre ré-assemblage et il devrait être traité comme un paquet pleinement ré-assemblé (c'est-à-dire, mettre à jour le prochain en-tête, ajuster la longueur de charge utile, retirer l'en-tête Fragment, etc.). Tous les autres fragments qui correspondent à ce paquet (c'est-à-dire, les mêmes adresses IPv6 de source, de destination, et identification de fragment) devraient être traités indépendamment.

Les conditions d'erreur suivantes peuvent survenir lors du ré-assemblage de paquets fragmentés :

- Si il n'y a pas assez de fragments reçus pour achever le ré-assemblage d'un paquet dans les 60 secondes de la réception du premier fragment arrivé de ce paquet, le ré-assemblage de ce paquet doit être abandonné et tous les fragments reçus pour ce paquet doivent être éliminés. Si le premier fragment (c'est-à-dire, celui qui a un décalage de fragment de zéro) a été reçu, un message ICMP Temps excédé -- temps de ré-assemblage de fragment excédé devrait être envoyé à la source de ce fragment.

- Si la longueur d'un fragment, comme déduite du champ Longueur de charge utile du paquet de fragment, n'est pas un multiple de 8 octets et si le fanion M de ce fragment est 1, ce fragment doit alors être éliminé et un message ICMP Problème de paramètre, code 0, devrait être envoyé à la source du fragment, pointant sur le champ Longueur de charge utile du paquet de fragment.

- Si la longueur et le décalage d'un fragment sont tels que la longueur de charge utile du paquet ré-assemblé à partir de ce fragment excéderait 65 535 octets, ce fragment doit alors être éliminé et un message ICMP Problème de paramètre, code 0, devrait être envoyé à la source du fragment, pointant sur le champ Décalage de fragment du paquet de fragment.

- Si le premier fragment n'inclut pas tous les en-têtes dans un en-tête de couche supérieure, ce fragment devrait alors être éliminé et un message ICMP Problème de paramètre, code 3, devrait être envoyé à la source du fragment, avec le champ Pointeur réglé à zéro.

- Si un des fragments ré-assemblé se chevauche avec un des autres fragments à ré-assembler pour le même paquet, le ré-assemblage de ce paquet doit être abandonné et tous les fragments qui ont été reçus pour ce paquet doivent être éliminés, et aucun message d'erreur ICMP ne devrait être envoyé.

On notera que les fragments peuvent être dupliqués dans le réseau. Au lieu de les traiter exactement comme des fragments

qui se chevauchent, une mise en œuvre peut choisir de détecter ces cas et d'éliminer les exacts dupliqués tout en conservant les autres fragments qui appartiennent au même paquet.

Les conditions suivantes ne sont pas supposées se produire fréquemment mais ne sont pas considérées comme des erreurs si elles se produisent :

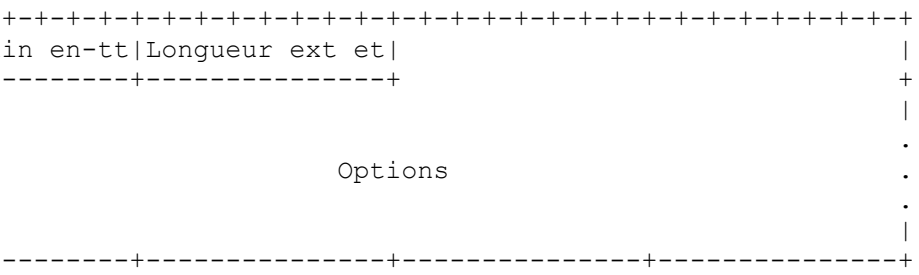
Le nombre et le contenu des en-têtes précédant l'en-tête Fragment des différents fragments du même paquet d'origine peuvent différer. Quels que soient les en-têtes présents, qui précèdent l'en-tête Fragment dans chaque paquet de fragment, ils sont traités quand les paquets arrivent, avant de mettre en file d'attente les fragments pour le ré-assemblage. Seuls les en-têtes dans le paquet de fragment de décalage zéro sont conservés dans le paquet ré-assemblé.

Les valeurs de prochain en-tête dans les en-têtes Fragment des différents fragments du même paquet d'origine peuvent différer. Seule la valeur provenant du paquet de fragment de décalage zéro est utilisée pour le ré-assemblage.

Les autres champs dans l'en-tête IPv6 peuvent aussi varier entre les fragments à ré-assembler. Les spécifications qui utilisent ces champs peuvent fournir des instructions supplémentaires si le mécanisme de base d'utilisation des valeurs provenant du fragment de décalage zéro n'est pas suffisant. Par exemple, le paragraphe 5.3 de la [RFC3168] décrit comment combiner les bits de notification explicite d'encombrement (ECN, *Explicit Congestion Notification*) provenant de différents fragments pour déduire les bits ECN du paquet ré-assemblé.

4.6 En-tête Options de destination

L'en-tête Options de destination est utilisé pour porter des informations facultatives qui n'ont besoin d'être examinées que par le ou les nœuds de destination d'un paquet. L'en-tête Options de destination est identifié par une valeur de prochain en-tête de 60 dans l'en-tête précédant immédiatement et a le format suivant :



Prochain en-tête : sélecteur de 8 bits. Identifie le type de l'en-tête qui suit immédiatement l'en-tête Options de destination. Utilise les mêmes valeurs que le champ Protocole IPv4 [IANA-PN].

Longueur d'extension d'en-tête : entier non signé de 8 bits. Longueur de l'en-tête Options de destination en unités de 8 octets, non inclus les huit premiers octets.

Options : champ de longueur variable, telle que l'en-tête Options de destination complet soit un multiple entier de 8 octets. Contient une ou plusieurs options codées en TLV, comme décrit au paragraphe 4.2.

Les seules options de destination définies dans ce document sont les options Pad1 et PadN spécifiées au paragraphe 4.2.

Noter qu'il y a deux façons possibles de coder des informations de destination facultatives dans un paquet IPv6 : soit comme option dans l'en-tête Options de destination, soit comme en-tête d'extension séparé. L'en-tête Fragment et l'en-tête Authentification sont des exemples de cette dernière approche. L'approche à utiliser dépend de l'action désirée d'un nœud de destination qui ne comprend pas les informations facultatives :

- Si l'action désirée est que le nœud de destination élimine le paquet et, seulement si l'adresse de destination du paquet n'est pas une adresse de diffusion groupée, envoie un message ICMP "Type non reconnu" à l'adresse de source du paquet, alors les informations peuvent être codées soit comme en-tête séparé, soit comme option dans l'en-tête Options de destination dont le type d'option a la valeur 11 dans ses 2 bits de plus fort poids. Le choix peut dépendre de facteurs tels que celui qui prend le moins d'octets, ou qui donne un meilleur alignement, ou l'analyse la plus efficace.
- Si une autre action est désirée, les informations doivent être codées comme option dans l'en-tête Options de destination dont le type d'option a la valeur 00, 01, ou 10 dans ses 2 bits de plus fort poids, spécifiant l'action désirée (voir au paragraphe 4.2).

4.7 Pas de prochain en-tête

La valeur 59 dans le champ Prochain en-tête d'un en-tête IPv6 ou de tout en-tête d'extension indique qu'il n'y a rien à la suite de cet en-tête. Si le champ Longueur de charge utile de l'en-tête IPv6 indique la présence d'octets après la fin d'un en-tête dont le champ Prochain en-tête contient 59, ces octets doivent être ignorés et passés inchangés si le paquet est transmis.

4.8 Définition de nouveaux en-têtes et options d'extension

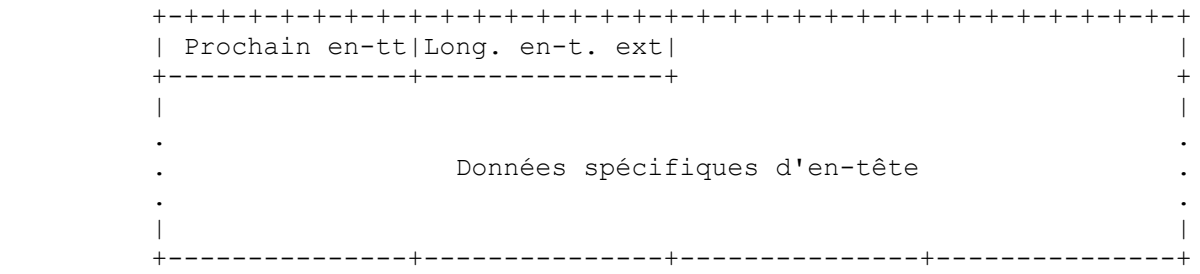
Définir de nouveaux en-têtes d'extension IPv6 n'est pas recommandé, sauf si il n'y a pas d'en-têtes d'extension IPv6 existants qui peuvent être utilisés en spécifiant une nouvelle option pour cet en-tête d'extension IPv6. Une proposition de spécification d'un nouvel en-tête d'extension IPv6 doit inclure une explication technique détaillée des raisons pour lesquelles un en-tête d'extension IPv6 existant ne peut pas être utilisé pour la nouvelle fonction désirée. Voir des informations supplémentaires dans la [RFC6564].

Note : de nouveaux en-têtes d'extension qui exigent un comportement bond par bond ne doivent pas être définis parce que, comme spécifié à la Section 4 de ce document, le seul en-tête d'extension qui ait un comportement bond par bond est l'en-tête Options bond par bond.

De nouvelles options bond par bond ne sont pas recommandées parce que les nœuds peuvent être configurés à ignorer l'en-tête Options bond par bond, à abandonner les paquets contenant un en-tête Options bond par bond, ou à allouer les paquets contenant un en-tête Options bond par bond à un chemin de traitement lent. Les concepteurs qui envisagent de définir de nouvelles options bond par bond doivent connaître de comportement probable. Il doit y avoir une justification très claire de la raison pour laquelle une option bond par bond est nécessaire avant qu'elle soit normalisée.

Au lieu de définir de nouveaux en-têtes d'extension, il est recommandé que les en-têtes d'options de destination soient utilisées pour porter des informations facultatives qui doivent n'être examinées que par le ou les nœuds de destination d'un paquet, parce que elles fournissent un meilleur traitement et la rétro compatibilité.

Si de nouveaux en-têtes d'extension sont définis, ils doivent utiliser le format suivant :



Prochain en-tête : sélecteur de 8 bits. Identifie le type d'en-tête qui suit immédiatement l'en-tête d'extension. Utilise les mêmes valeurs que le champ Protocole IPv4 [IANA-PN].

Longueur d'en-tête d'extension : entier non signé de 8 bits. Longueur de l'en-tête Options de destination en unités de 8 octets, non inclus les 8 premiers octets.

Données spécifique d'en-tête : champ de longueur variable. Champs spécifiques de l'en-tête d'extension.

5. Problèmes de taille de paquet

IPv6 exige que chaque liaison dans l'Internet ait une MTU de 1280 octets ou plus. Ceci est appelé la MTU de liaison IPv6 minimum. Sur chaque liaison qui ne peut pas porter un paquet de 1280 octets en une seule fois, une fragmentation et un réassemblage spécifique de la liaison doivent être fournis à une couche en dessous de IPv6.

Les liaisons qui ont une MTU configurable (par exemple, les liaisons PPP [RFC1661]) doivent être configurées à avoir une MTU d'au moins 1280 octets ; il est recommandé qu'elles soient configurées avec une MTU de 1500 octets ou plus, pour s'accommoder de possibles encapsulations (c'est-à-dire, tunnelage) sans subir de fragmentation de couche IPv6.

À partir de chaque liaison à laquelle un nœud est directement rattaché, le nœud doit être capable d'accepter des paquets

aussi gros que la MTU de cette liaison.

Il est fortement recommandé que les nœuds IPv6 mettent en œuvre la découverte de la MTU de chemin [RFC8201], afin de découvrir et tirer parti des MTU de chemin supérieures à 1280 octets. Cependant, une mise en œuvre minimale de IPv6 (par exemple, dans une ROM d'amorçage) peut simplement se limiter à envoyer des paquets de pas plus de 1280 octets, et omettre la mise en œuvre de la découverte de la MTU de chemin.

Afin d'envoyer un paquet plus gros que la MTU d'un chemin, un nœud peut utiliser l'en-tête Fragment IPv6 pour fragmenter le paquet à la source et le faire ré-assembler à sa ou ses destinations. Cependant, l'utilisation d'une telle fragmentation est déconseillée dans toute application qui est capable d'ajuster ses paquets pour qu'ils tiennent dans la MTU de chemin mesurée (c'est-à-dire, moins de 1280 octets).

Un nœud doit être capable d'accepter un paquet fragmenté qui, après ré-assemblage, fait jusqu'à 1500 octets. Il est permis à un nœud d'accepter des paquets fragmentés qui ré-assemblés font plus de 1500 octets. Un protocole ou application de couche supérieure qui dépend de la fragmentation IPv6 pour envoyer des paquets plus grands que la MTU d'un chemin ne devrait pas envoyer de paquets de plus de 1500 octets sauf si il a l'assurance que la destination est capable de ré-assembler des paquets de cette plus grande taille.

6. Étiquettes de flux

Le champ de 20 bits Étiquette de flux dans l'en-tête IPv6 est utilisé par une source pour étiqueter les séquences de paquets qui vont être traitées dans le réseau comme un seul flux. La définition actuelle de l'étiquette de flux IPv6 se trouve dans la [RFC6437].

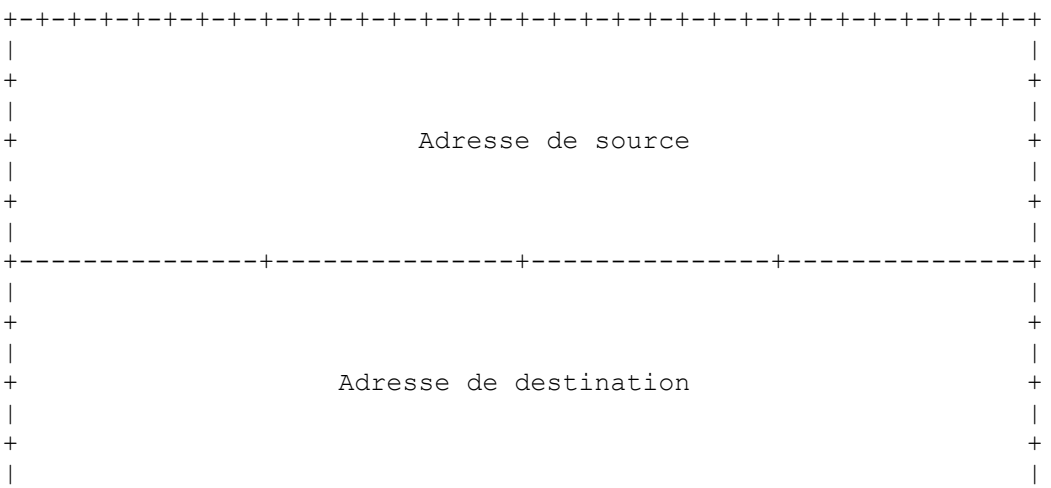
7. Classes de trafic

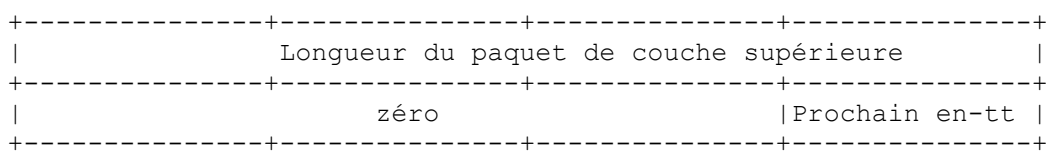
Le champ de 8 bits Classe de trafic dans l'en-tête IPv6 est utilisé par le réseau pour la gestion du trafic. La valeur des bits de classe de trafic dans un paquet ou fragment reçu peut être différente de la valeur envoyée par la source du paquet. L'utilisation actuelle du champ Classe de trafic pour les services différenciés et la notification explicite d'encombrement sont spécifiées dans les [RFC2474] et [RFC3168].

8. Questions de protocole de couche supérieure

8.1 Somme de contrôle de couche supérieure

Tout protocole de transport ou autre de couche supérieure qui inclut les adresses provenant de l'en-tête IP dans son calcul de somme de contrôle doit être modifié pour l'utiliser sur IPv6, afin d'inclure les adresses IPv6 de 128 bits au lieu des adresses IPv4 de 32 bits. En particulier, l'illustration suivante montre le "pseudo en-tête" TCP et UDP pour IPv6 :





- Si le paquet IPv6 contient un en-tête Acheminement, l'adresse de destination utilisée dans le pseudo en-tête est celle de la destination finale. Au nœud d'origine, cette adresse va être dans le dernier élément de l'en-tête Acheminement ; chez le ou les receveurs, cette adresse va être dans le champ Adresse de destination de l'en-tête IPv6.

- La valeur de prochain en-tête dans le pseudo en-tête identifie le protocole de couche supérieure (par exemple, 6 pour TCP ou 17 pour UDP). Elle va différer de la valeur de prochain en-tête dans l'en-tête IPv6 si il y a des en-têtes d'extension entre l'en-tête IPv6 et l'en-tête de couche supérieure.

- La longueur de paquet de couche supérieure dans le pseudo en-tête est la longueur de l'en-tête de couche supérieure et des données (par exemple, en-tête TCP plus données TCP). Certains protocoles de couche supérieure portent leurs propres informations de longueur (par exemple, le champ Longueur dans l'en-tête UDP) ; pour de tels protocoles, c'est la longueur utilisée dans le pseudo-en-tête. D'autres protocoles (comme TCP) ne portent pas leurs propres informations de longueur, et dans ce cas, la longueur utilisée dans le pseudo en-tête est la longueur de charge utile provenant de l'en-tête IPv6, moins la longueur de tout en-tête d'extension présent entre l'en-tête IPv6 et l'en-tête de couche supérieure.

- À la différence de IPv4, le comportement par défaut quand des paquets UDP sont générés par un nœud IPv6 est que la somme de contrôle UDP n'est pas facultative. C'est-à-dire que chaque fois qu'il génère un paquet UDP, un nœud IPv6 doit calculer une somme de contrôle UDP sur le paquet et le pseudo en-tête, et, si ce calcul donne un résultat de zéro, il doit être changé en FFFF hexadécimal pour être placé dans l'en-tête UDP. Les receveurs IPv6 doivent éliminer les paquets UDP qui contiennent une somme de contrôle de zéro et devraient enregistrer l'erreur dans le journal des événements.

- Par exception au comportement par défaut, les protocoles qui utilisent UDP comme encapsulation de tunnel peuvent permettre un mode de somme de contrôle zéro pour un accès spécifique (ou ensemble d'accès) pour l'envoi et/ou la réception. Tout nœud qui met en œuvre un mode de somme de contrôle zéro doit respecter les exigences de la [RFC6936] "Déclaration d’applicabilité pour l’utilisation de datagrammes UDP IPv6 avec somme de contrôle à zéro".

La version IPv6 de ICMP [RFC4443] inclut le pseudo en-tête ci-dessus dans son calcul de somme de contrôle ; c'est donc un changement par rapport à la version IPv4 de ICMP, qui n'inclut pas le pseudo en-tête dans sa somme de contrôle. La raison du changement est de protéger ICMP contre la mauvaise livraison ou la corruption de ces champs d'en-tête IPv6 dont il dépend, et qui, à la différence de IPv4, ne sont pas couverts par une somme de contrôle de couche internet. Le champ Prochain en-tête dans le pseudo en-tête pour ICMP contient la valeur 58, qui identifie la version IPv6 de ICMP.

8.2 Durée de vie maximum de paquet

À la différence de IPv4, les nœuds IPv6 ne sont pas obligés d'appliquer une durée de vie maximum de paquet. C'est la raison pour laquelle le champ IPv4 "Durée-de-vie" a été renommé "Limite de bonds" dans IPv6. En pratique, très peu, si il en est, de mises en œuvre IPv4 se conforment à l'exigence de limitation de la durée de vie de paquet, de sorte qu'il n'y a pas de changement en pratique. Tout protocole de couche supérieure qui s'appuie sur la couche internet (IPv4 ou IPv6) pour limiter la durée de vie d'un paquet devrait être mis à niveau pour fournir son propre mécanisme de détection et d'élimination des paquets obsolètes.

8.3 Taille maximum de charge utile de couche supérieure

Quand il calcule la taille maximum de charge utile disponible pour les données de couche supérieure, un protocole de couche supérieure doit tenir compte de la plus grande taille de l'en-tête IPv6 par rapport à l'en-tête IPv4. Par exemple, dans IPv4, l'option Taille maximum de segment (MSS, *Maximum Segment Size*) de TCP est calculée comme la taille maximum de paquet (valeur par défaut ou valeur apprise par la découverte de la MTU de chemin) moins 40 octets (20 octets pour la longueur minimale d'en-tête IPv4 et 20 octets pour la longueur minimale d'en-tête TCP). Quand on utilise TCP sur IPv6, la MSS doit être calculée comme taille maximum de paquet moins 60 octets, parce que la longueur minimum d'en-tête IPv6 (c'est-à-dire, un en-tête IPv6 sans en-tête d'extension) est plus longue de 20 octets que la longueur minimum de l'en-tête IPv4.

8.4 Réponse aux paquets qui portent des en-têtes d'acheminement

Quand un protocole de couche supérieure envoie un ou plusieurs paquets en réponse à un paquet reçu qui incluait un en-tête Acheminement, le ou les paquets de réponse ne doivent pas inclure d'en-tête Acheminement qui a été automatiquement déduit en "inversant" l'en-tête Acheminement SAUF si l'intégrité et l'authenticité de l'adresse de source et l'en-tête Acheminement reçus ont été vérifiés (par exemple, via l'utilisation d'un en-tête Authentification dans le paquet reçu). En d'autres termes, seuls les sortes de paquets suivantes sont permises en réponse à un paquet reçu qui porte un en-tête Acheminement :

- Les paquets de réponse qui ne portent pas d'en-tête Acheminement.
- Les paquets de réponse qui portent des en-têtes Acheminement qui NE sont PAS déduits en inversant l'en-tête Acheminement du paquet reçu (par exemple, un en-tête Acheminement fourni par la configuration locale).
- Les paquets de réponse qui portent des en-têtes Acheminement qui ont été déduit en inversant l'en-tête Acheminement du paquet reçu SI ET SEULEMENT SI l'intégrité et l'authenticité de l'adresse de source et l'en-tête Acheminement du paquet reçu ont été vérifiés par le répondant.

9. Considérations relatives à l'IANA

La RFC 2460 est référencée dans un certain nombre de registres de l'IANA. Cela inclut :

- Paramètres du Protocole Internet version 6 (IPv6) [IANA-6P]
- Numéros alloués du protocole Internet [IANA-PN]
- Identifiants de réseau ONC RPC (netids) [IANA-NI]
- Identifiants de protocole de couche réseau (NLPID) intéressants [IANA-NL]
- Registres de protocoles [IANA-PR]

L'IANA a mis à jour ces références pour pointer sur le présent document.

10. Considérations sur la sécurité

IPv6, du point de vue du format de base et de la transmission des paquets, a des propriétés de sécurité similaires à IPv4. Ces problèmes de sécurité incluent :

- L'espionnage, où des éléments sur le chemin peuvent observer tout le paquet (incluant le contenu et les métadonnées) de chaque datagramme IPv6.
- La répétition, où l'attaquant enregistre une séquence de paquets hors réseau et les répète à la partie qui les a reçus à l'origine.
- L'insertion de paquets, où l'attaquant falsifie un paquet avec un ensemble choisi de propriétés et les injecte dans le réseau.
- La suppression de paquets, où l'attaquant retire un paquet du réseau.
- La modification de paquet, où l'attaquant retire un paquet du réseau, le modifie, et le réinjecte dans le réseau.
- Les attaques par interposition, où l'attaquant subvertit le flux de communication afin de se faire passer pour l'envoyeur au receveur et pour le receveur à l'envoyeur.
- Les attaques de déni de service (DoS) où l'attaquant envoie de grandes quantités de trafic légitime à une destination pour la submerger.

Les paquets IPv6 peuvent être protégés de l'espionnage, de la répétition, de l'insertion de paquet, de la modification de paquet, et des attaques par interposition en utilisant "L'architecture de sécurité pour le protocole Internet" [RFC4301]. De plus, des protocoles de couche supérieure comme la sécurité de la couche transport (TLS, *Transport Layer Security*) ou Secure Shell (SSH) peuvent être utilisés pour protéger le trafic de couche application par dessus IPv6.

Il n'y a aucun mécanisme pour protéger contre les attaques de DoS. La défense contre ce type d'attaques sort du domaine d'application de la présente spécification.

Les adresses IPv6 sont significativement plus grandes que les adresses IPv4 rendant plus difficile l'examen de l'espace d'adresses à travers l'Internet et même sur une seule liaison réseau (par exemple, réseau de zone locale). Voir plus d'informations dans la [RFC7707].

Les adresses IPv6 de nœuds sont supposées être plus visibles sur l'Internet qu'avec IPv4 car l'utilisation de la technique de traduction d'adresse est réduite. Cela crée des problèmes de confidentialité supplémentaires comme de rendre plus facile de distinguer les points d'extrémité. Voir plus d'informations dans la [RFC7721].

La conception de l'architecture d'en-tête d'extension IPv6, bien qu'elle apporte beaucoup de souplesse, crée aussi de nouveaux défis pour la sécurité. Comme on le note ci-dessous, les problèmes relatifs à l'en-tête d'extension de fragment ont été résolus, mais il est clair que pour tout nouvel en-tête d'extension qui sera conçu à l'avenir, les implications pour la sécurité devront être examinées avec soin, et cela doit inclure comment le nouvel en-tête d'extension fonctionne avec les en-têtes d'extension existants. Voir plus d'informations dans la [RFC7045].

Cette version de la spécification IPv6 résout un certain nombre de problèmes de sécurité qui ont été trouvés dans la version précédente [RFC2460] de la spécification IPv6. Cela inclut :

- De réviser le texte pour traiter le cas des fragments qui sont des datagrammes entiers (c'est-à-dire, à la fois le champ Décalage de fragment et le fanion M sont à zéro). Si ils sont reçus, ils devraient être traités comme un paquet ré-assemblé. Tous les autres fragments qui correspondent devraient être traités indépendamment. Le processus de création de fragment a été modifié pour ne pas créer de fragments sur un datagramme complet (avec le champ Décalage de fragment et le fanion M à zéro). Voir plus d'informations dans les [RFC6946] et [RFC8021].
- De supprimer le paragraphe de la Section 5 qui exigeait d'inclure un en-tête Fragment aux paquets sortants si un message ICMP Paquet trop gros rapportait une MTU de prochain bond de moins de 1280. Voir plus d'informations dans la [RFC6946].
- De changer le texte pour exiger que les nœuds IPv6 ne créent pas de fragments en chevauchement. Aussi, quand un datagramme IPv6 est ré-assemblé, si un ou plusieurs de ses fragments constituants sont déterminés être des fragments en chevauchement, le datagramme entier (et tous les fragments constituants) doivent être éliminés en silence. Cela inclut la précision qu'aucun message d'erreur ICMP ne devrait être envoyé si des fragments en chevauchement sont reçus. Voir plus d'informations dans la [RFC5722].
- De réviser le texte pour exiger que tous les en-têtes depuis le premier en-tête de couche supérieure soient dans le premier fragment. Voir plus d'informations dans la [RFC7112].
- D'incorporer les mises à jour de la [RFC5095] et de la [RFC5871] pour retirer la description du type d'en-tête Acheminement 0 (RH0) et que les lignes directrices d'allocations pour les en-têtes Acheminement sont spécifiées dans la RFC 5871, et retirer RH0 de la liste des en-têtes d'extension exigés.

Les problèmes de sécurité relatifs aux autres parties de IPv6 incluant l'adressage, ICMPv6, la découverte de la MTU de chemin, etc., sont discutés dans les spécifications appropriées.

11. Références

11.1 Références normatives

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

[RFC2474] K. Nichols, S. Blake, F. Baker et D. Black, "Définition du champ Services différenciés (DS) dans les en-têtes IPv4 et IPv6", décembre 1998. *(P.S. ; MàJ par* RFC3168, RFC3260, RFC8436)

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

[RFC4291] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", février 2006. (*MàJ par* 5952 et 6052, 8064) *(D.S.)*

[RFC4443] A. Conta et autres, "Spécification du protocole de message de contrôle Internet (ICMPv6) pour la version 6 du protocole Internet (IPv6)", mars 2006. (*Remplace* RFC2463 ; *MàJ* RFC2780 ; *MàJ par* RFC4884 ; *D.S.)*

[RFC6437] S. Amante, B. Carpenter, S. Jiang, J. Rajahalme, "Spécification de l'étiquette de flux IPv6", novembre 2011. (*Remplace la* RFC3697) (*MàJ les* RFC2205, RFC2460) *(P.S.)*

11.2 Références pour information

[Err2541] RFC Errata, Erratum ID 2541, RFC 2460.

[Err4279] RFC Errata, Erratum ID 4279, RFC 2460.

[Err4657] RFC Errata, Erratum ID 4657, RFC 2460.

[Err4662] RFC Errata, Erratum ID 4662, RFC 2460.

[IANA-6P] IANA, "Internet Protocol Version 6 (IPv6) Parameters", <https://www.iana.org/assignments/ipv6-parameters >.

[IANA-EH] IANA, "IPv6 Extension Header Types", <https://www.iana.org/assignments/ipv6-parameters >.

[IANA-NI] IANA, "ONC RPC Network Identifiers (netids)", <https://www.iana.org/assignments/rpc-netids >.

[IANA-NL] IANA, "Network Layer Protocol Identifiers (NLPID) of Interest", <https://www.iana.org/assignments/nlpids>.

[IANA-PN] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers >.

[IANA-PR] IANA, "Protocol Registries", <https://www.iana.org/protocols >.

[IANA-RH] IANA, "Routing Types", <https://www.iana.org/assignments/ipv6-parameters >.

[RFC1661] W. Simpson, éditeur, "Protocole point à point (PPP)", STD 51, juillet 1994. *(MàJ par la* RFC2153)

[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*)*

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. *(P.S.) (Remplace la* RFC2401)

[RFC4302] S. Kent, "En-tête d'authentification IP", décembre 2005. *(P.S.)*

[RFC4303] S. Kent, "Encapsulation de charge utile de sécurité dans IP (ESP)", décembre 2005. (*Remplace* RFC2406) *(P.S.)*

[RFC5095] J. Abley, P. Savola, G. Neville-Neil "En-têtes d'acheminement de type 0 déconseillés dans IPv6**",** décembre 2007. *(P.S.)*

[RFC5722] S. Krishnan, "Traitement des fragments IPv6 en chevauchement", décembre 2009. *(P. S.)*

[RFC5871] J. Arkko, S. Bradner, "Lignes directrices de l'allocation par l'IANA de l'en-tête d'acheminement IPv6", mai 2010. (*MàJ* RFC2460). *(P. S.)*

[RFC6564] S. Krishnan et autres, "Format uniforme pour les en-têtes d'extension IPv6", avril 2012. (*MàJ la* RFC2460) *(P.S.)*

[RFC6936] G. Fairhurst, M. Westerlund, "Déclaration d'applicabilité pour l'utilisation de datagrammes UDP IPv6 avec somme de contrôle à zéro", avril 2013. *(P.S.)*

[RFC6946] F. Gont, "Traitement des fragments IPv6 "Atomic"", mai 2013. (*MàJ* RFC2460, 5722) *(P.S.)*

[RFC7045] B. Carpenter, S. Jiang, "Transmission et traitement des en-têtes d'extension IPv6", décembre 2013. (*MàJ* RFC2460, RFC2780) *(P.S.)*

[RFC7112] F. Gont, V. Manral, R. Bonica, "Implications des chaînes d'en-tête IPv6 surdimensionnées", janvier 2014. (*MàJ* RFC2460) *(P.S.)*

[RFC7707] F. Gont, T. Chown, "Reconnaissance de réseau dans les réseaux IPv6", mars 2016. *(Information)*

[RFC7721] A. Cooper, F. Gent, D. Thaler, "Considérations de sécurité et de confidentialité pour les mécanismes de génération d'adresse IPv6", mars 2016. *(Information)*

[RFC7739] F. Gont, "Implications pour la sécurité de valeurs d'identification de fragment prévisibles", février 2016.

(Info)

[RFC8021] F. Gont, W. Liu, T. Andersen, "La génération de fragments IPv6 atomiques est dommageable", janvier 2017. *(Info)*

[RFC8201] J. McCann, et autres, "Découverte de la MTU de chemin pour IPv6", juillet 2017. STD 87. *(Remplace* RFC1981)

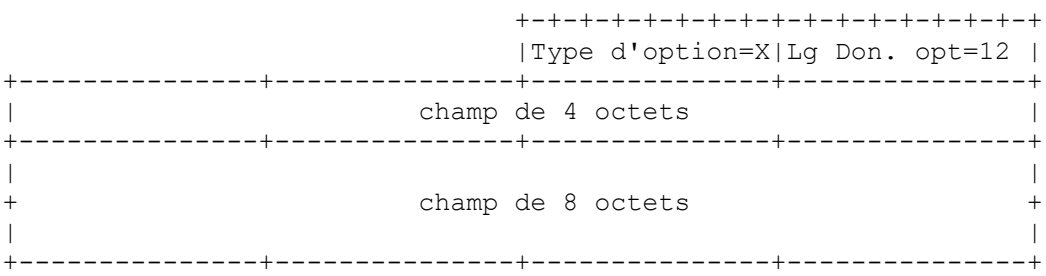
Appendice A. Lignes directrices pour le formatage des options

Cet appendice donne des indications sur la façon de disposer les champs lors de la conception de nouvelles options à utiliser dans les en-têtes Options bond par bond ou les en-têtes d'options de destination, comme décrit au paragraphe 4.2. Ces lignes directrices se fondent sur les hypothèses suivantes :

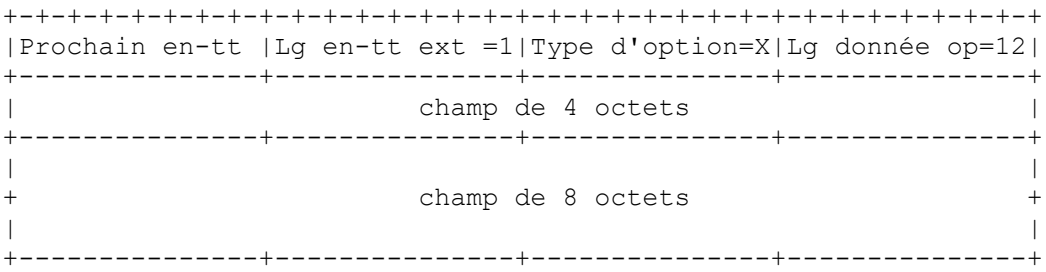
- o Une caractéristique désirable est que tout champ multi-octets dans la zone des données d'option d'une option soient alignés sur leur limite naturelle, c'est-à-dire que les champs de n octets devraient être placés à un multiple entier de n octets du début de l'en-tête d'option bond par bond ou Destination, pour n = 1, 2, 4, ou 8.
- o Une autre caractéristique désirable est que l'en-tête d'option bond par bond ou Destination prenne aussi peu d'espace que possible, sous réserve de l'exigence que l'en-tête soit un multiple entier de 8 octets.
- o On peut supposer que, quand des en-têtes de support d'option sont présents, ils portent un très petit nombre d'options, généralement seulement une.

Ces hypothèses suggèrent l'approche suivante pour disposer les champs d'une option : ordonner les champs du plus petit au plus grand, sans bourrage interne, puis déduire l'exigence d'alignement pour l'option entière sur la base de l'exigence d'alignement du plus grand champ (jusqu'à un alignement maximum de 8 octets). Cette approche est illustrée dans les exemples qui suivent :

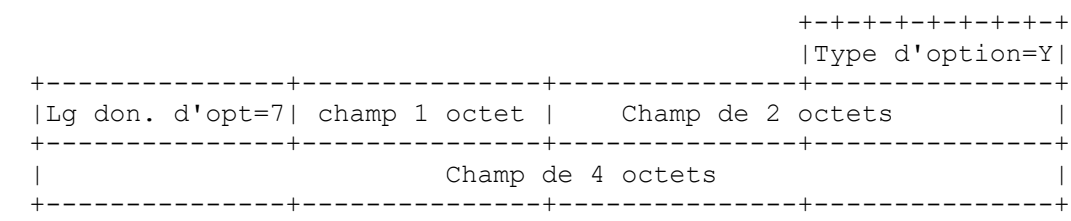
Exemple 1
Si une option X exige deux champs de données, une de longueur 8 octets et une de longueur 4 octets, elle serait disposée comme suit :



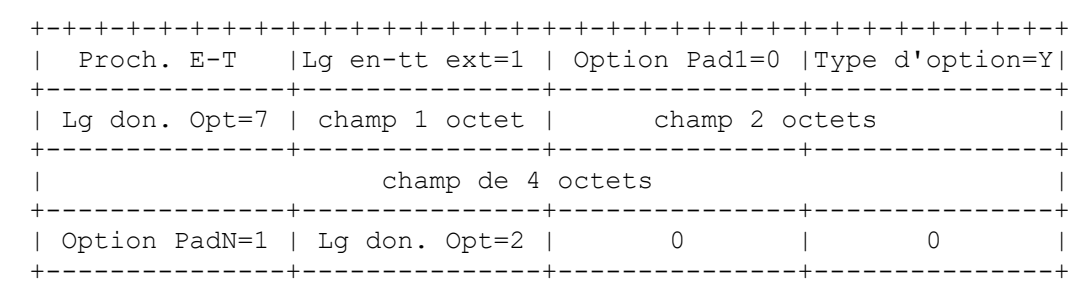
Son exigence d'alignement est 8n+2, pour assurer que le champ de 8 octets commence à un décalage multiple de 8 à partir du début de l'en-tête qui l'enclot. Un en-tête complet d'options bond par bond ou Destination contenant cette option ressemblerait à ceci :



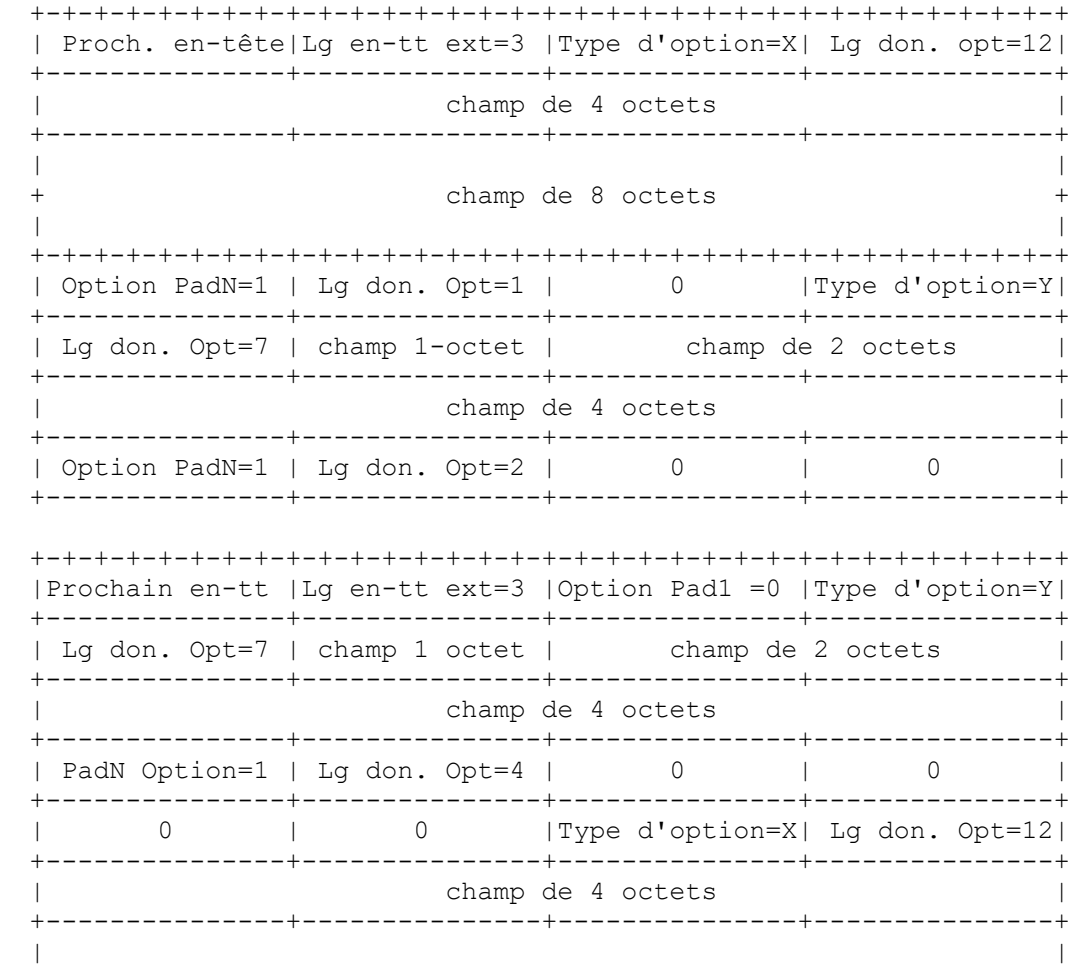
Exemple 2
Si une option Y exige trois champs de données, un de 4 octets, un de 2 octets, et u d'1 octet, il va se présenter comme suit :

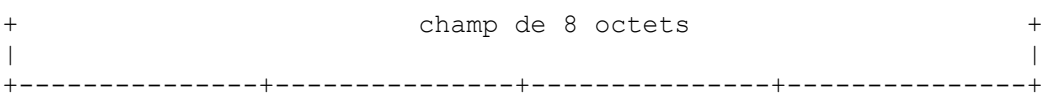


Son exigence d'alignement est 4n+3, pour assurer que le champ de 4 octets commence à un décalage multiple de 4 à partir du début de l'en-tête qui l'enclot. Un en-tête complet d'options bond par bond ou Destination contenant cette option ressemblerait à ceci :



Exemple 3
Un en-tête Options bond par bond ou Destination contenant les deux options X et Y des exemples 1 et 2 aurait un des deux formats suivants, selon l'option qui apparaît en premier :





Appendice B. Changements depuis la RFC 2460

Le présent mémoire a apporté les changements suivants par rapport à la RFC 2460.

- Suppression de IP de prochaine génération du résumé.
- Ajout de texte à la Section 1 sur l'ordre de transmission des données qui est le même que dans IPv4 comme défini dans la RFC 791.
- Précisé le texte de la Section 3 sur la décrémentation de la limite de bonds.
- Précisé que les en-têtes d'extension (sauf l'en-tête Options bond par bond) ne sont pas traités, insérés, ou supprimés par un nœud le long du chemin de livraison d'un paquet.
- Changé l'exigence pour l'en-tête Options bond par bond en un "peut", et ajout d'une note pour indiquer ce qui est attendu concernant l'en-tête Options bond par bond.
- Ajout d'un paragraphe à la Section 4 pour préciser comment les en-têtes d'extension sont numérotés et lesquels sont des en-têtes de couche supérieure.
- Ajout d'une référence à la fin de la Section 4 au registre IANA "Types d'en-tête d'extension IPv6".
- Incorporation des mises à jour des RFC 5095 et 5871 pour retirer la description de RH0, que les lignes directrices d'allocations pour les en-têtes d'acheminement sont spécifiées dans la RFC 5871, et retrait de RH0 de la liste des en-têtes d'extension exigés.
- Révision du paragraphe 4.5 sur la fragmentation IPv6 sur la base des mises à jour des RFC 5722, 6946, 7112, et 8021. Cela inclut :
  - La révision du texte pour traiter le cas des fragments qui sont des datagrammes complets (c'est-à-dire, le champ Décalage de fragment et le fanion M sont tous deux à zéro). Si il en est reçus, ils devraient être traités comme un paquet ré-assemblé Tous les autres fragments qui correspondent devraient être traités indépendamment. Le processus de création de fragment révisé a été modifié pour ne pas créer de fragments qui soient un datagramme complet (le champ Décalage de fragment et le fanion M sont à zéro).
  - De changer le texte pour exiger que les nœuds IPv6 ne créent pas de fragments en chevauchement. Aussi, lors du ré-assemblage d'un datagramme IPv6, si un ou plusieurs de ses fragments constituants sont déterminés comme étant en recouvrement, le datagramme entier (et tous les fragments constituants) doit être éliminé en silence. Cela inclut la précision qu'aucun message d'erreur ICMP ne devrait être envoyé si des fragments en chevauchement sont reçus.
  - De réviser le texte pour exiger que tous les en-têtes jusqu'au premier en-tête de couche supérieure soient dans le premier fragment. Cela change le texte qui décrit comment les paquets sont fragmentés et ré-assemblés et ajout d'un nouveau cas d'erreur.
  - Ajout d'un texte au processus de traitement de l'en-tête Fragment de fragments dupliqués exacts.
  - Mise à jour du texte de l'en-tête Fragmentation pour corriger l'inclusion d'un en-tête Authentification (AH) et noter le cas de Pas de prochain en-tête.
  - Changer la terminologie dans le paragraphe "en-tête Fragment" de "en-tête non fragmentable" en "en-tête par fragment".
  - Suppression du paragraphe de la Section 5 qui exigeait d'inclure un en-tête Fragment sur les paquets sortants si un message ICMP "Paquet trop gros" rapporte une MTU de prochain bond de moins de 1280.
  - Changer le texte pour préciser la restriction de MTU et les restrictions à 8 octets, et noter la restriction sur les en-têtes dans le premier fragment.
- Au paragraphe 4.5, ajout d'une précision pour noter que des champs dans l'en-tête IPv6 peuvent aussi varier selon les fragments ré-assemblés, et que d'autres spécifications peuvent donner des instructions supplémentaires sur la façon dont ils devraient être ré-assemblés. Voir, par exemple, le paragraphe 5.3 de la [RFC3168].
- Incorporé la mise à jour de la RFC 6564 d'ajouter un nouveau paragraphe 4.8 décrivant les recommandations pour définir de nouveaux en-têtes d'extension et options.
- Ajout de texte à la Section 5 pour définir la "MTU de liaison IPv6 minimum".
- Simplification du texte de la Section 6 sur les étiquettes de flux et retrait de ce qui était l'Appendice A ("Sémantique et usage du champ Étiquette de flux") ; pointe à la place sur les spécifications actuelles du champ Étiquette de flux IPv6 dans la [RFC6437] et le champ Classe de trafic dans les [RFC2474] et [RFC3168].
- Incorporation de la mise à jour de la RFC 6935 ("Sommes de contrôle IPv6 et UDP pour les paquets tunnelés") à la Section 8. Ajout d'une exception au comportement par défaut pour le traitement des paquets UDP avec des sommes de contrôle à zéro pour les tunnels.
- Ajout d'une instruction à la Section 9, "Considérations relatives à l'IANA", pour changer les références à la RFC 2460 en références au présent document.
- Révision et expansion de la Section 10, "Considérations sur la sécurité".

- o Ajout d'un paragraphe à la section des remerciement pour remercier les auteurs des documents de mise à jour.
- o Mise à jour des références aux versions actuelles et répartition des références en normatives et pour information.
- o Fait les changements pour résoudre les errata sur la RFC 2460. Ce sont :

Erratum ID 2541 [Err2541] : il note que la RFC 2460 ne met pas à jour la RFC 2205 quand la longueur de l'étiquette de flux a été passée de 24 à 20 bits à partir de la RFC 1883. Ce problème a été résolu dans la RFC 6437 où l'étiquette de flux est définie. Cette spécification fait maintenant référence à la RFC 6437. Aucun changement n'est requis.

Erratum ID 4279 [Err4279] : il note que la spécification ne traite pas le cas d'un nœud qui reçoit un paquet avec une limite de bond de zéro. Ceci est traité à la Section 3 de la présente spécification.

Erratum ID 4657 [Err4657] : il propose du texte disant que les en-têtes d'extension ne doivent jamais être insérés par un nœud autre que la source du paquet. Ceci a été réglé à la Section 4, "En-têtes d'extension IPv6".

Erratum ID 4662 [Err4662] : il propose un texte disant que les en-têtes d'extension, à une exception, ne sont pas examinés, traités, modifiés, insérés, ou supprimés par tout nœud le long du chemin de livraison d'un paquet. Ceci a été réglé à la Section 4, "En-têtes d'extension IPv6".

Erratum ID 2843 : Cet erratum est marqué "Rejeté". Aucun change n'a été fait.

Remerciements

Les auteurs remercient chaleureusement de leurs nombreuses suggestions utiles les membres du groupe de travail IPng, du groupe de recherche sur les protocoles de bout en bout, et la communauté de l'Internet dans son ensemble.

Les auteurs souhaitent aussi remercier les auteurs des RFC de mise à jour qui ont été incorporées dans ce document pour faire passer la spécification IPv6 au statut de norme. Ce sont Joe Abley, Shane Amante, Jari Arkko, Manav Bhatia, Ronald P. Bonica, Scott Bradner, Brian Carpenter, P.F. Chimento, Marshall Eubanks, Fernando Gont, James Hoagland, Sheng Jiang, Erik Kline, Suresh Krishnan, Vishwas Manral, George Neville-Neil, Jarno Rajahalme, Pekka Savola, Magnus Westerlund, et James Woodyatt.

Adresse des auteurs

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, CA 94070
USA
mèl : bob.hinden@gmail.com

Stephen E. Deering
Vancouver, British Columbia
Canada