Groupe de Travail sur les Réseaux
Requête pour Commentaires : RFC 2460 - FR
Remplace : 1883
Catégorie : Standard

S. Deering
Cisco
R. Hinden
Nokia
traduction N. Jourdan, Ecole polytechnique
de l'université de Nantes

Internet Protocol, Version 6 (IPv6)-Specification
le Protocole Internet Version 6-Spécification

Note du traducteur

Ce document est une traduction intégrale non officielle, mais sans ajouts (excepté cette note), sans commentaires et sans omissions, du RFC 2460 tel qu'édité par ses auteurs, spécifiant le protocole Internet standard IPv6. L'auteur de cette traduction décline toute responsabilité sur l'utilisation de ce document et/ou sur d'éventuelles erreurs de traduction.

Concernant les droits du traducteur : le traducteur renonce à ses droits sur la reproduction de ce document si l'ensemble de ces conditions est respecté : les reproductions doivent être **complètes** (contenant cette note), **d'un seul tenant** (un seul fichier ou un ensemble de pages physiquement reliées), **sans aucune modification** du contenu et **réalisées à partir de la dernière version** de ce document disponible gratuitement sur le site mentionné ci-après.

Dernière version du document

http://abcdrfc.free.fr/

Adresse du traducteur

M. Nicolas Jourdan
3, Impasse du Clos des Pins
30870 CLARENSAC
France
E-mail : njourdan@free.fr

Statut de ce Document

Ce document est une spécification d'un protocole Internet reconnu et approuvé comme un standard par la communauté Internet ; il propose une discussion sur le protocole et des suggestions pour des améliorations. Merci de se reporter à la dernière version des " Standards Officiels des Protocoles Internet " (STD 1 - " Internet Official Protocol Standards ") pour connaître l'état de standardisation et le statut de ce protocole. La distribution du RFC-2460 original (en anglais) n'est pas limitée.

Information sur les droits d'auteur

Résumé

Ce document spécifie la version 6 du Protocole Internet (IPv6), parfois dénommé aussi IP Nouvelle Génération ou IPng (IP Next Generation).

Table des matières

1. Introduction

IP version 6 (IPv6) est une nouvelle version du Protocole Internet, désigné comme le successeur de IP version 4 (IPv4) [RFC-791]. Les changements entre IPv4 et IPv6 se répartissent en première approche dans les catégories suivantes :

*** Augmentation des possibilités d'adressage**

IPv6 augmente la taille des adresses IP de 32 bits à 128 bits, pour supporter plusieurs niveaux de hiérarchies d'adressages, un bien plus grand nombre de nœuds adressables et une auto-configuration plus simple des adresses. Une plus grande souplesse de configuration du multicasting est

obtenue grâce à l'ajout d'un champ " groupe " (" scope ") aux adresses de type multicast. Ensuite, un nouveau type d'adresse est défini : les adresses de type " diffusion à tous " (" anycast adress "), utilisées pour envoyer un paquet à toutes les entités d'un groupe de nœuds.

*** Simplification du format de l'en-tête**

Certains champs de l'en-tête IPv4 ont été enlevés ou rendus optionnels, pour réduire dans les situations classiques le coût (en ressources de traitement) de la gestion des paquets et pour limiter le surcoût en bande passante de l'en-tête IPv6.

*** Support amélioré des options et des extensions futures**

Des changements dans la façon dont les options de l'en-tête IP sont encodés permettent une transmission (forwarding) plus efficace, des limites moins strictes sur la longueur des options et une plus grande flexibilité dans l'introduction par la suite de nouvelles options.

*** Fonctionnalité d'étiquetage de flux d'informations**

Une nouvelle fonctionnalité est ajoutée pour étiqueter des paquets appartenant à des " flux " d'informations particuliers pour lesquels l'émetteur demande une gestion spéciale, comme un service " sans perte d'information " ou un service " temps réel ".

*** Fonctionnalité d'authentification et de confidentialité**

Des extensions pour gérer l'authentification, l'intégrité des données ou une (optionnelle) confidentialité des données sont spécifiées par IPv6.

Ce document spécifie l'en-tête IPv6 de base, les en-têtes d'extension initialement définis et les options d'IPv6. Il discute aussi des problèmes de longueur des paquets, de la sémantique des labels de flux d'informations, de la sémantique des classes de trafic, et des conséquences d'IPv6 sur les protocoles de couche supérieure. Le format et la sémantique des adresses IPv6 sont spécifiés séparément dans [ADDRARCH]. La version IPv6 d'ICMP, que toutes les implémentations d'IPv6 doivent inclure, est spécifiée dans [ICMPv6].

2. Terminologie

nœud (node)
un système qui implémente IPv6.

routeur (router)
un nœud qui transmet (forward) des paquets qui ne lui sont pas explicitement adressés. [Voir note ci-après].

hôte (host)
tout nœud qui n'est pas un routeur. [Voir note ci-après].

couche supérieure (upper layer)
une couche de protocole immédiatement au-dessus d'IPv6. Des exemples sont les protocoles de transport tels que TCP et UDP, les protocoles de contrôle tels qu'ICMP, les protocoles de routage tels qu'OSPF et les protocoles inter-réseaux ou des couches inférieures passant à l'intérieur de tunnels (i.e., qui sont encapsulés à l'intérieur de - " tunneled " over) IPv6 tels qu'IPX, AppleTalk ou IPv6 lui-même.

lien (link)

un médium ou un canal de communication par lequel des nœuds peuvent communiquer à partir de la couche de lien, i.e., la couche immédiatement au-dessous d'IPv6. Des exemples sont les réseaux Ethernet (simples ou avec des ponts) ; les liaisons PPP ; les réseaux X.25, Frame Relay ou ATM ; et les " tunnels " de couche inter-réseaux (ou supérieure) tels que les tunnels à travers IPv4 ou IPv6 lui-même.

voisinage (neighbors)

des nœuds reliés par un même lien.

interface (interface)

un dispositif du nœud qui le relie au lien.

adresse (address)

un identificateur de la couche IPv6 pour une interface ou un ensemble d'interfaces.

paquet (packet)

un en-tête IPv6 avec sa " charge utile " (ce qu'il transporte).

MTU de lien (link MTU)

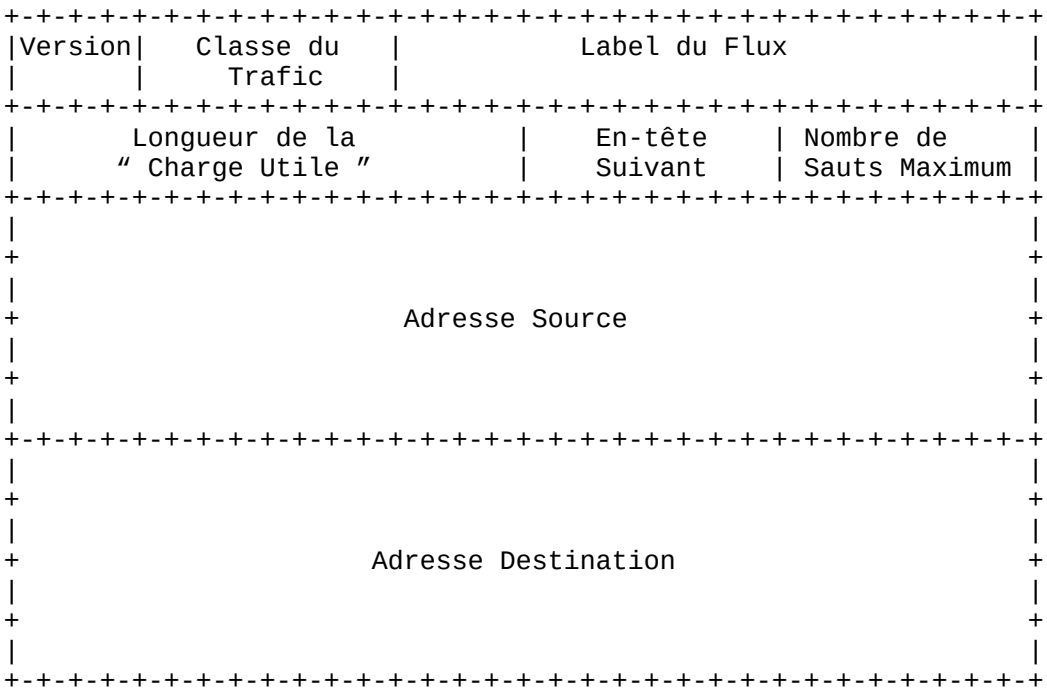
l'unité maximum de transmission (Maximum Transmission Unit), i.e., la taille maximale en octets d'un paquet qui peut être transmis sur le lien.

MTU de chemin (path MTU)

le plus petit MTU de lien de l'ensemble des liens constituant le chemin entre le nœud source et le nœud destination.

Note : il est impossible (quoique cette situation soit rare) pour un système ayant plusieurs interfaces, en ce qui concerne les paquets qui ne lui sont pas destinés, d'être configuré à la fois pour transmettre ceux qui arrivent d'un ensemble de ses interfaces (mais pas de toutes) et pour se débarrasser de ceux qui arrivent de ses autres interfaces. Un tel système doit satisfaire aux exigences du protocole pour les routeurs quand il reçoit des paquets à partir des premières interfaces et interagit par celles-ci avec le voisinage (transmission – forwarding). Il doit satisfaire aux exigences du protocole pour les hôtes quand il reçoit des paquets à partir des secondes interfaces et interagit par celles-ci avec le voisinage (non-transmission – non-forwarding).

3. Format de l'en-tête IPv6



Version (Version)
numéro de version du Protocole Internet (= 6) sur 4 bits.

Classe du Trafic (Traffic Class)
champ sur 8 bits relatif à la classe de trafic. [Voir section 7].

Label du Flux (Flow Label)
Label sur 20 bits relatif au flux d'information. [Voir section 6].

Longueur de la "Charge Utile" (Payload Length)
Entier non-signé sur 16 bits. Longueur en octets de la charge utile, i.e., le reste du paquet qui suit cet en-tête IPv6. (Il faut noter que tous les en-têtes d'extension [section 4] présents sont considérés comme faisant partis de la charge utile, i.e., inclus dans le décompte de la longueur.)

En-tête Suivant (Next Header)
Sélecteur sur 8 bits. Identifie le type de l'en-tête suivant immédiatement l'en-tête IPv6. Utilise les mêmes valeurs que le champ " protocole " d'IPv4 [RFC-1700 et suivants].

Nombre de Sauts Maximum (Hop Limit)
Entier non-signé sur 8 bits. Décrémenté de 1 par chaque nœud que le paquet traverse. Le paquet est éliminé si le Nombre de Sauts Maximum arrive à zéro.

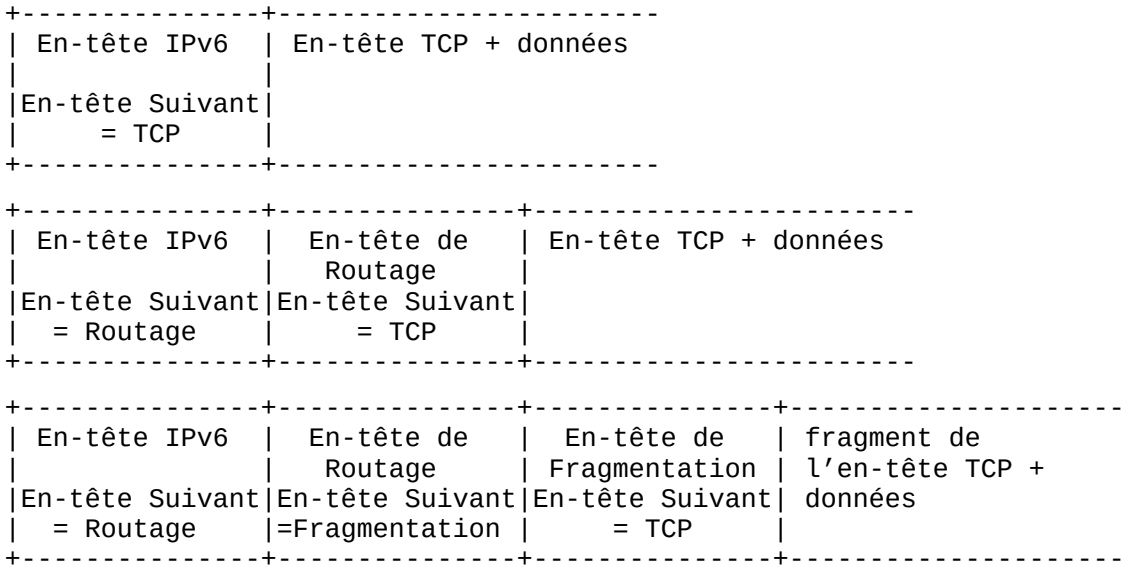
Adresse Source (Source Address)
adresse sur 128 bits de l'expéditeur initial du paquet. Voir [ADDRARCH].

Adresse Destination (Destination Address)

adresse sur 128 bits du destinataire projeté du paquet (qui peut ne pas être le destinataire ultime, si un en-tête de routage est présent). Voir [ADDRARCH] et [section 4.4].

4. En-têtes IPv6 d'extension

Avec IPv6, les informations optionnelles de la couche inter-réseaux sont encodées dans des en-têtes séparés qui peuvent être placés entre l'en-tête IPv6 et l'en-tête de couche supérieure d'un paquet. Il y a un petit nombre de tels en-têtes d'extension, chacun identifié par une valeur distincte d'" En-tête Suivant ". Comme présenté dans ces exemples, un paquet IPv6 peut transporter zéro, un ou plusieurs en-têtes d'extension, chacun identifié par le champ " En-tête Suivant " de l'en-tête précédent :



A part une exception, les en-têtes d'extension ne sont pas examinés ou traités par un quelconque nœud le long du chemin emprunté par le paquet (packet's delivery path), jusqu'à ce que le paquet atteigne le nœud (ou l'ensemble de nœuds, dans le cas du multicasting) identifié par le champ " Adresse Destination " de l'en-tête IPv6. Là, le démultiplexage normal à partir du champ " En-tête Suivant " de l'en-tête IPv6 appelle le module pour gérer le premier en-tête d'extension ou l'en-tête de plus haut niveau s'il n'y a pas d'en-tête d'extension. Le contenu et la sémantique de chaque en-tête d'extension déterminent si oui ou non il faut poursuivre l'analyse sur l'en-tête suivant. De plus, les en-têtes d'extension doivent être analysés dans l'ordre exact où ils apparaissent dans le paquet ; un destinataire ne doit pas, par exemple, commencer par chercher un type particulier d'en-tête d'extension et traiter cet en-tête avant tous ceux qui le précèdent.

L'exception dont fait allusion le précédent paragraphe est l'en-tête des options sauts après sauts (Hop-by-Hop Options Header), qui transporte les options qui doivent être examinées et traitées par chaque nœud le long du chemin emprunté par le paquet, incluant les nœuds source et destination. L'en-tête des options sauts après sauts, quand il est présent, doit immédiatement suivre l'en-tête IPv6. Sa présence est indiquée par la valeur zéro dans le champ " En-tête Suivant " de l'en-tête IPv6.

Si, lors du traitement des en-têtes, un nœud atteint une valeur d'" En-tête Suivant " non reconnue à l'intérieur de l'en-tête qu'il traite, il devrait éliminer le paquet et envoyer un message ICMP " Problème de Paramètre " (ICMP Parameter Problem message) à la source de ce paquet, avec un code ICMP de 1 (" Rencontre d'un type d'En-tête Suivant inconnu " - " unrecognized Next Header type encountered ") et le champ " Pointeur " d'ICMP contenant la position de la valeur inconnue à l'intérieur du paquet original. Il

devrait être fait la même chose si un nœud rencontre une valeur d'“ En-tête Suivant ” à zéro dans tout en-tête autre qu'un en-tête IPv6.

Tout en-tête d'extension a une longueur multiple de 8 octets, dans le but de maintenir un alignement sur 8 octets des en-têtes suivants. Les champs sur plusieurs octets à l'intérieur de chaque en-tête d'extension sont alignés suivant leur propre longueur, i.e., les champs de n octets de long sont placés tous les multiples entiers de n octets à partir du début de l'en-tête, pour n=1, 2, 4 ou 8.

Une implémentation complète de IPv6 inclut l'implémentation des en-têtes d'extension suivants :

- en-tête des options sauts après sauts
- en-tête de routage (type 0)
- en-tête de fragmentation
- en-tête des options de destination
- en-tête d'authentification
- en-tête d'encapsulation de charge utile sécurisée

Les quatre premiers sont spécifiés dans ce document ; les deux derniers sont spécifiés dans les [RFC-2402] et [RFC-2406], respectivement.

4.1. Ordre des en-têtes d'extension

Quand il y a plus d'un en-tête d'extension dans un même paquet, il est recommandé que ces en-têtes apparaissent dans l'ordre suivant :

- en-tête IPv6
- en-tête des options sauts après sauts
- en-tête des options de destination (note 1)
- en-tête de routage
- en-tête de fragmentation
- en-tête d'authentification (note 2)
- en-tête d'encapsulation de charge utile sécurisée (note 2)
- en-tête des options de destination (note 3)
- en-tête de couche supérieure

note 1 : pour les options qui seront traitées par la destination première qui apparaît dans le champ Adresse Destination d'IPv6 ainsi que par les destinations ultérieures, listées dans l'en-tête de routage.

note 2 : des recommandations complémentaires concernant l'ordre relatif des en-têtes d'authentification et d'encapsulation de charge utile sécurisée sont données dans [RFC-2406].

note 3 : pour les options qui doivent être traitées uniquement par le destinataire final du paquet.

Chaque en-tête d'extension devrait apparaître au plus une fois, excepté l'en-tête des options de destination qui devrait apparaître au plus deux fois (une fois avant un en-tête de routage et une fois avant l'en-tête de couche supérieure).

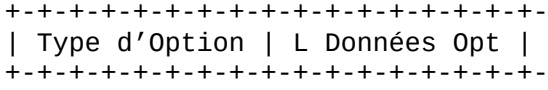
Si l'en-tête de couche supérieure est un autre en-tête IPv6 (dans le cas d'un tunnel IPv6 à travers IPv6, i.e. dans le cas où IPv6 est encapsulé dans IPv6), il peut être suivi par ses propres en-têtes d'extension, qui sont sujets séparément aux mêmes recommandations sur leur ordre d'apparition.

Si et quand d'autres en-tête d'extensions sont définis, les contraintes sur leur ordre d'apparition par rapport aux en-têtes listés précédemment doivent être spécifiées.

Les nœuds IPv6 doivent accepter et tenter de traiter les en-têtes d'extension quel que soit l'ordre dans lequel ils apparaissent et quel que soit le nombre de fois où ils apparaissent dans un même paquet, excepté pour l'en-tête des options sauts après sauts qui doit obligatoirement suivre immédiatement et uniquement un en-tête IPv6. Néanmoins, il est fortement conseillé que les sources de paquets IPv6 adhèrent à l'ordre recommandé précédemment jusqu'à ce que et même si des spécifications ultérieures révisent cette recommandation.

4.2. Options

Deux des en-têtes d'extension actuellement définis – l'en-tête des options sauts après sauts et l'en-tête des options de destination – transportent un nombre variable d'" options " encodées type – longueur - valeur (TLV), suivant le format ci-après :



Type d'Option (Option Type)

Identificateur sur 8 bits du type d'option.

L Données Opt (Opt Data Len)

Entier 8 bits non signé, longueur du champ Données de l'Option de cette option, en octets.

Données de l'Option (Option Data)

Champ de longueur variable, données spécifiques au Type d'Option.

La séquence d'options à l'intérieur d'un en-tête doit être traitée strictement dans l'ordre où apparaissent ces options dans l'en-tête ; un récepteur ne doit pas, par exemple, parcourir l'en-tête à la recherche d'un type particulier d'option et traiter cette option avant d'avoir traité toutes celles qui la précèdent.

Les identificateurs de Type d'Option sont encodés en interne de telle sorte que leurs deux bits de poids fort spécifient ce qui doit être fait si le nœud qui traite l'en-tête IPv6 ne reconnaît pas le Type d'Option :

- 00 – ne pas tenir compte de cette option et continuer à traiter l'en-tête
- 01 – éliminer le paquet

- 10 – éliminer le paquet et, suivant si l'Adresse Destination du paquet est une adresse multicast ou non, envoyer un message ICMP Problème de Paramètre, Code 2 à l'Adresse Source du paquet, pointant sur le Type d'Option non reconnu.

- 11 – éliminer le paquet et, seulement si l'Adresse Destination du paquet n'est pas une adresse multicast, envoyer un message ICMP Problème de Paramètre, Code 2 à l'Adresse Source du paquet, pointant sur le Type d'Option non reconnu.

Le troisième bit de poids fort du Type d'Option spécifie si les Données de l'Option peuvent changer en cours de route vers le destinataire final du paquet. Quand un en-tête d'authentification est présent dans le paquet, le champ Données de l'Option complet de toutes les options dont les données peuvent changer en cours de route doit être traité comme une zone d'octets à zéro lors du calcul ou de la vérification de la valeur d'authentification du paquet.

Signification du troisième bit de poids fort :

- 0 – Les Données de l'Option ne changent pas en cours de route

- 1 – Les Données de l'Option changent en cours de route

Les trois bits de poids fort décrits précédemment sont à traiter comme partie intégrante du Type d'Option et non indépendamment du Type d'Option. Aussi, une option particulière est identifiée par un Type d'Option complet sur 8 bits et non juste par ses 5 bits de poids faible.

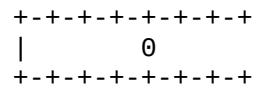
Le même espace de numérotation des Types d'Option est utilisé à la fois pour l'en-tête des options sauts après sauts et pour l'en-tête des options de destination. Cependant, la spécification d'un type particulier d'option peut restreindre son utilisation à un seul de ces deux en-têtes.

Les options individuelles peuvent avoir des exigences spécifiques d'alignement, pour assurer que des valeurs sur plusieurs octets à l'intérieur des champs de Données de l'Option tombent sur des frontières naturelles. L'exigence d'alignement d'une option est spécifiée en utilisant la notation xn+y, signifiant 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 n fois 2 octets à partir du début de l'en-tête
8n+2	signifie tout décalage de n fois 8 octets à partir du début de l'en-tête, plus 2 octets

Il y a deux options de bourrage (padding options) qui sont utilisées lorsqu'il est nécessaire d'aligner les options suivantes ou pour faire en sorte que l'en-tête ait une longueur multiple de 8 octets.

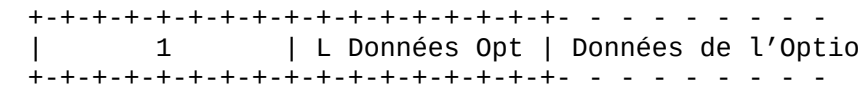
Option Pad1 (exigence d'alignement : aucune)



note ! Le format de l'option Pad1 est un cas spécial – il n'a pas de champ de longueur et de données.

L'option Pad1 est utilisée pour insérer un octet de bourrage à l'intérieur des zones d'options d'un en-tête. Si plus d'un octet de bourrage est nécessaire, l'option PadN, décrite ci-après, devrait être utilisée à la place de plusieurs options Pad1.

Option PadN (exigence d'alignement : aucune)

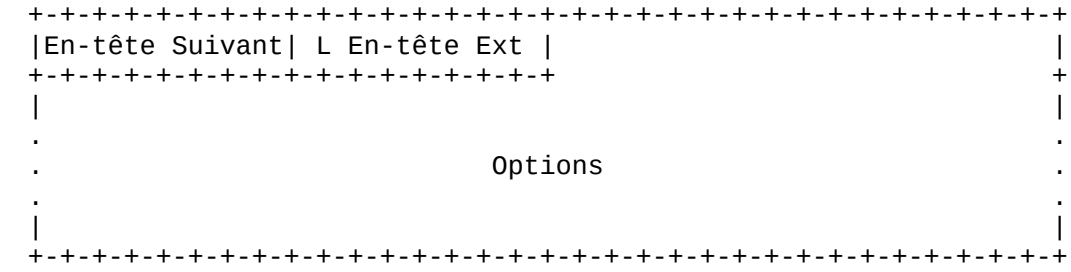


L'option PadN est utilisée pour insérer au moins deux octets de bourrage à l'intérieur des zones d'options d'un en-tête. Pour N octets de bourrage, le champ L Données Opt contient la valeur N-2 et les Données de l'Option consistent en N-2 octets à zéro.

L'Appendice B présente des directives de formatage pour de nouvelles options.

4.3. En-tête des options sauts après sauts

L'en-tête des options sauts après sauts est utilisé pour transporter des informations optionnelles qui doivent être examinées par chaque nœud le long du chemin emprunté par le paquet. L'en-tête des options sauts après sauts est identifié par une valeur d'En-tête Suivant à 0 dans l'en-tête IPv6 et a le format suivant :



En-tête (Next Header)

Sélecteur sur 8 bits. Identifie le type de l'en-tête suivant immédiatement l'en-tête des options sauts après sauts. Utilise les mêmes valeurs que le champ " protocole " d'IPv4 [RFC-1700 et suivants].

L En-tête Ext (Hdr Ext Len)

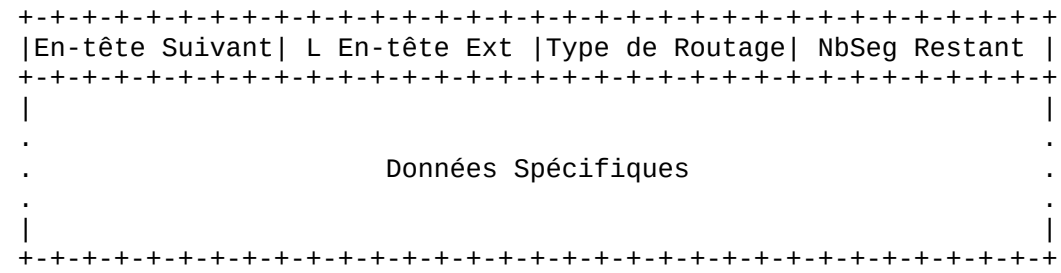
Entier 8 bits non signé. Longueur de l'en-tête des options sauts après sauts en mots de 8 octets, sans compter les 8 premiers octets.

Options (Options)

Champ de longueur variable, telle que l'en-tête des options sauts par sauts complet soit un entier multiple de 8 octets. Contient au moins une option encodée TLV, comme décrit dans la section 4.2.

4.4. En-tête de routage

L'en-tête de routage est utilisé par une source IPv6 pour lister au moins un nœud intermédiaire à " aller voir " sur le chemin emprunté par le paquet vers la destination. Cette fonction est très similaire aux options de " source vague " (" Loose Source ") ou d'enregistrement de chemin (" Record Route ") d'IPv4. L'en-tête de routage est identifié par une valeur d'En-tête Suivant à 43 dans l'en-tête le précédent immédiatement et a le format suivant :



En-tête Suivant (Next Header)

Sélecteur sur 8 bits. Identifie le type de l'en-tête suivant immédiatement l'en-tête de routage. Utilise les mêmes valeurs que le champ " protocole " d'IPv4 [RFC-1700 et suivants].

L En-tête Ext (Hdr Ext Len)

Entier 8 bits non signé. Longueur de l'en-tête de routage en mots de 8 octets, sans compter les 8 premiers octets.

Type de Routage (Routing Type)

Identificateur sur 8 bits de la variante particulière de l'en-tête de routage.

NbSeg Restant (Segments Left)

Entier 8 bits non signé. Nombre de segments de chemin restant, i.e., nombre de nœuds intermédiaires listés explicitement qu'il reste à traverser avant d'atteindre la destination finale.

Données Spécifiques (type-specific data)

Champ de longueur variable, de format déterminé par le Type de Routage, et de longueur telle que l'en-tête de routage complet soit un multiple entier de 8 octets.

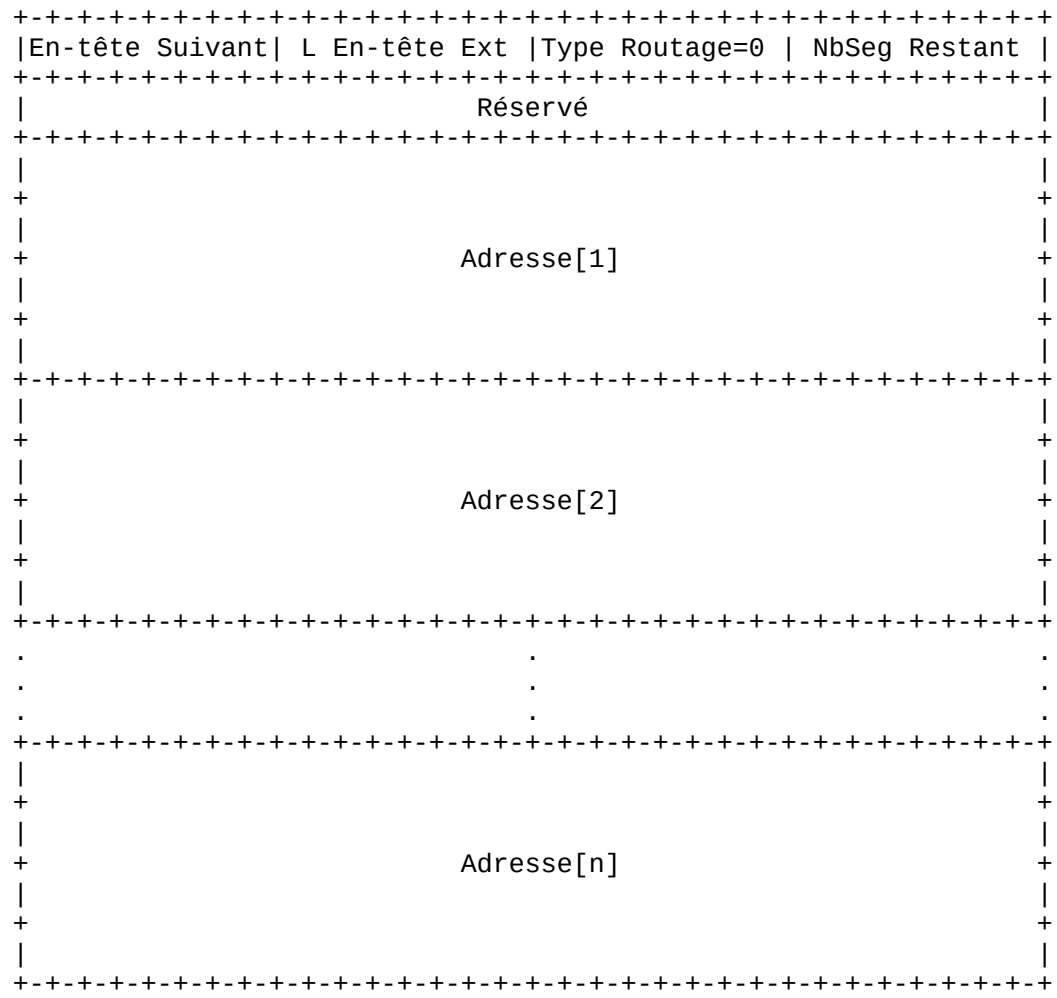
Si, au cours du traitement d'un paquet reçu, un nœud rencontre un en-tête de routage avec une valeur de Type de Routage inconnue, le comportement exigé du nœud dépend de la valeur du champ NbSeg Restant, comme ci-après :

Si le NbSeg Restant est zéro, le nœud doit ignorer l'en-tête de routage et procéder au traitement de l'en-tête suivant dans le paquet, dont le type est identifié par le champ En-tête Suivant dans l'en-tête de routage.

Si le NbSeg Restant est différent de zéro, le nœud doit éliminer le paquet et envoyer un message ICMP Problème de Paramètre, Code 0, pointant sur le Type de Routage inconnu, à l'adresse source du paquet.

Si, après le traitement de l'en-tête de routage d'un paquet reçu, un nœud intermédiaire détermine que le paquet est à transférer sur un lien dont le MTU de lien est inférieur à la taille du paquet, le nœud doit éliminer le paquet et envoyer un message ICMP Paquet Trop Gros (Packet Too Big) à l'adresse source du paquet.

L'en-tête de routage de Type 0 a le format suivant :



En-tête Suivant (Next Header)

Sélecteur sur 8 bits. Identifie le type de l'en-tête suivant immédiatement l'en-tête de routage. Utilise les mêmes valeurs que le champ " protocole " d'IPv4 [RFC-1700 et suivants].

L En-tête Ext (Hdr Ext Len)

Entier 8 bits non signé. Longueur de l'en-tête de routage en mots de 8 octets, sans compter les 8 premiers octets. Pour l'en-tête de routage de Type 0, L En-tête Ext est égal à deux fois le nombre d'adresses dans l'en-tête.

Type de Routage (Routing Type)

0.

NbSeg Restant (Segments Left)

Entier 8 bits non signé. Nombre de segments de chemin restant, i.e., nombre de nœuds intermédiaires listés explicitement qu'il reste à traverser avant d'atteindre la destination finale.

Réservé (Reserved)

Champ réservé de 32 bits. Initialisé à zéro pour la transmission ; ignoré en réception.

Adresse[1..n] (Address[1..n])

Vecteur d'adresses de 128 bits, numérotées de 1 à n.

Les adresses multicast ne doivent pas apparaître dans un en-tête de routage de type 0 ou dans le champ Adresse Destination IPv6 d'un paquet transportant un en-tête de routage de type 0.

Un en-tête de routage n'est pas examiné ou traité tant qu'il n'atteint pas le nœud identifié par le champ Adresse Destination de l'en-tête IPv6. Dans ce nœud, l'expédition du champ En-tête Suivant de l'en-tête le précédant immédiatement provoque l'appel du module gérant l'en-tête de routage, qui, dans le cas d'un Type de Routage à 0, exécute l'algorithme suivant :

```
si NbSeg Restant = 0 {
   procéder au traitement de l'en-tête suivant dans le paquet,
   dont le type est identifié par le champ En-tête Suivant de
   l'en-tête de routage
}
sinon si L En-tête Ext est impaire {
   envoyer un message ICMP Problème de Paramètre, Code 0, pointant sur
   le champ L En-tête Ext, à l'Adresse Source et éliminer le paquet
}
sinon {
   calculer n, le nombre d'adresses dans l'en-tête de routage, en
   divisant L En-tête Ext par 2

   si NbSeg Restant est plus grand que n {
      envoyer un message ICMP Problème de Paramètre, Code 0, pointant
      sur le champ NbSeg Restant, à l'Adresse Source et éliminer le
      paquet
   }
   sinon {
      décrémenter NbSeg Restant de 1 ;
      calculer i, l'index de la prochaine adresse du vecteur
      d'adresses des nœuds à aller voir, en soustrayant NbSeg Restant
      de n

      si Adresse[i] ou Adresse Destination IPv6 sont de type multicast {
         éliminer le paquet
      }
      sinon {
         échanger Adresse Destination IPv6 et Adresse[i]

         si le Nombre de Sauts Maximum IPv6 est inférieur ou égal à 1 {
            envoyer un message ICMP Délai Dépassé – Nombre de Sauts
            Maximum Dépassé durant l'acheminement (Time Exceeded – Hop
            Limit Exceeded in Transit) à l'Adresse Source et éliminer
            le paquet
         }
         sinon {
            décrémenter le Nombre de Sauts Maximum de 1

            renvoyer le paquet au module IPv6 pour le transmettre à la
            nouvelle destination
         }
      }
   }
}
```

Comme exemple des effets de l'algorithme précédent, considérons le cas où un nœud source S envoie un paquet au nœud destination D, en utilisant un en-tête de routage afin que le paquet soit routé par les nœuds intermédiaires I1, I2 et I3. Les valeurs des

champs utiles de l'en-tête IPv6 et de l'en-tête de routage sur chaque segment du chemin emprunté par le paquet seraient les suivantes :

- Quand le paquet voyage de S à I1 :

Adresse Source = S	L En-tête Ext = 6
Adresse Destination = I1	NbSeg Restant = 3
	Adresse[1] = I2
	Adresse[2] = I3
	Adresse[3] = D

- Quand le paquet voyage de I1 à I2 :

Adresse Source = S	L En-tête Ext = 6
Adresse Destination = I2	NbSeg Restant = 2
	Adresse[1] = I1
	Adresse[2] = I3
	Adresse[3] = D

- Quand le paquet voyage de I2 à I3 :

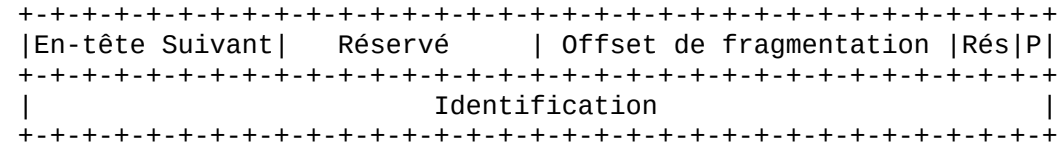
Adresse Source = S	L En-tête Ext = 6
Adresse Destination = I3	NbSeg Restant = 1
	Adresse[1] = I1
	Adresse[2] = I2
	Adresse[3] = D

- Quand le paquet voyage de I3 à D :

Adresse Source = S	L En-tête Ext = 6
Adresse Destination = D	NbSeg Restant = 0
	Adresse[1] = I1
	Adresse[2] = I2
	Adresse[3] = I3

4.5. En-tête de fragmentation

L'en-tête de fragmentation est utilisé par une source IPv6 pour envoyer un paquet plus large que celui qui tiendrait dans le MTU de chemin vers la destination. (Note : contrairement à IPv4, la fragmentation dans IPv6 n'est réalisée que par les nœuds sources, non par les routeurs le long d'un chemin emprunté par le paquet – voir section 5.) L'en-tête de fragmentation est identifié par une valeur d'En-tête Suivant de 44 dans l'en-tête le précédent immédiatement. Il a le format suivant :



En-tête Suivant (Next Header)

Sélecteur sur 8 bits. Identifie le type d'en-tête initial de la Partie Fragmentable du paquet original (définie ci-après). Utilise les mêmes valeurs que le champ Protocole d'IPv4 [RFC-1700 et suivants].

Réservé (Reserved)

Champ réservé de 8 bits. Initialisé à zéro pour la transmission ; ignoré en réception.

Offset de fragmentation (Fragment Offset)

Entier 13 bits non signé. L'offset des données suivant cet en-tête, en nombre de mots de 8 octets, par rapport au début de la Partie Fragmentable (Fragmentable Part) du paquet original.

Rés (Res)

Champ réservé de 2 bits. Initialisé à zéro pour la transmission ; ignoré en réception.

Drapeau P (M flag)

1 = plus de fragments ; 0 = dernier fragment

Identification (Identification)

32 bits. Voir description ci-après.

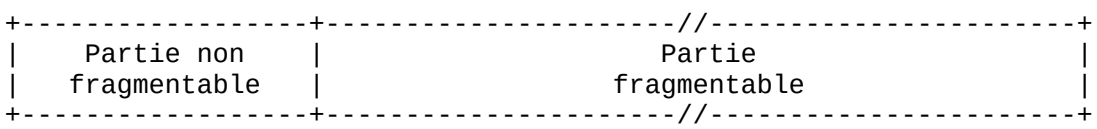
Pour pouvoir envoyer un paquet qui est trop large pour tenir dans le MTU de chemin de sa destination, un nœud source peut diviser le paquet en fragments et envoyer chaque fragment comme un paquet distinct. Ces paquets distincts seront rassemblés ensuite par le destinataire.

Pour chaque paquet devant être fragmenté, le nœud source génère une valeur d'Identification. L'Identification doit être différente de tout autre identification de paquet fragmenté envoyé récemment* avec les mêmes Adresse Source et Adresse Destination. Si un en-tête de routage est présent, l'Adresse Destination concernée est celle de la destination finale.

* " récemment " signifie à l'intérieur du temps de vie maximum vraisemblable, incluant le temps de transit de la source à la destination et le temps passé à attendre l'assemblage des autres fragments du même paquet. Cependant, il n'est pas exigé qu'un nœud source connaisse la durée de vie maximum d'un paquet. Plutôt, il est supposé que cette exigence peut être satisfaite en entretenant la valeur d'identification comme un simple compteur 32 bits en boucle, incrémenté chaque fois qu'un paquet doit être fragmenté. C'est un choix d'implémentation d'entretenir un unique compteur pour le nœud ou plusieurs compteurs, par exemple un pour chaque adresse source possible du nœud ou un pour chaque combinaison (adresse source, adresse destination) active.

Le paquet initial, long et non fragmenté est dénommé le " paquet original " et est considéré comme constitué de deux parties, comme illustré ci-après :

Paquet original :

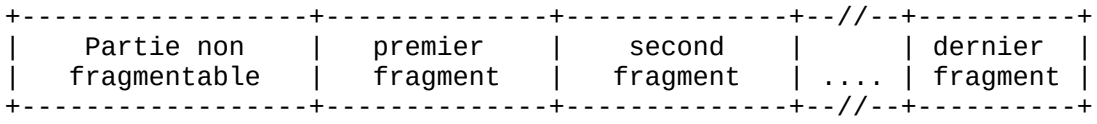


La partie non fragmentable (Unfragmentable Part) est constituée de l'en-tête IPv6 et de tous les en-têtes d'extension qui doivent être traités par des nœuds en cours de route vers la destination, il s'agit de tous les en-têtes jusqu'à et incluant l'en-tête de routage s'il est présent, sinon l'en-tête des options sauts après sauts s'il est présent, sinon pas d'en-tête d'extension.

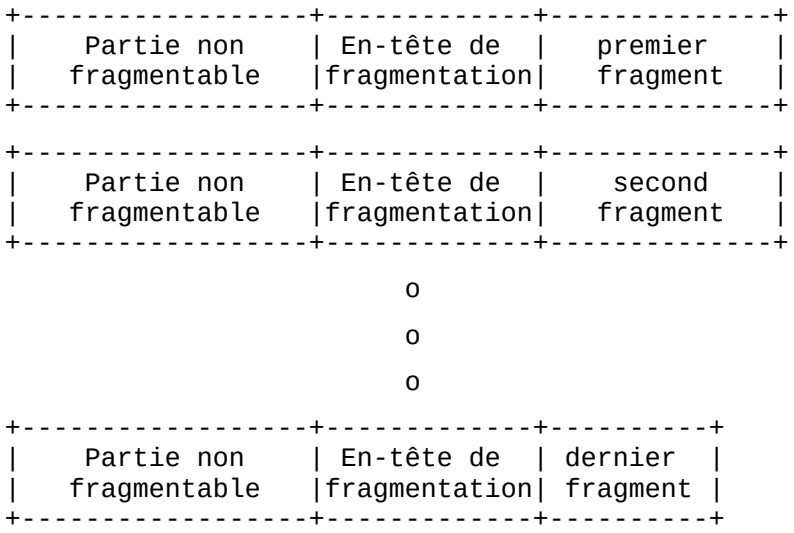
La partie fragmentable est constituée du reste du paquet, il s'agit de tout en-tête d'extension qui ne nécessite que le traitement du (des) nœud(s) destination finale ainsi que l'en-tête et les données de couche supérieure.

La partie fragmentable du paquet original est divisée en fragments, chacun, excepté peut-être le dernier (“ le plus à droite ”), étant un multiple entier de mots de 8 octets. Les fragments sont transmis dans des “ paquets fragmentés ” (“ fragment packets ”) comme illustré ci-après.

Paquet original :



Paquets fragmentés :



Chaque paquet fragmenté est composé de :

(1) La partie non fragmentable du paquet original, avec la Longueur de la Charge Utile de l’en-tête IPv6 original changé en la longueur de ce fragment uniquement (en excluant la longueur de l’en-tête IPv6 lui-même), et le champ d’En-tête Suivant du dernier en-tête de la partie non fragmentable changé en 44.

(2) Un en-tête de fragmentation contenant :

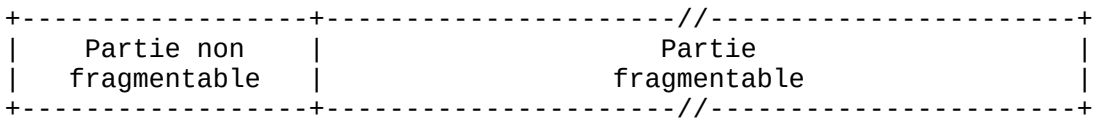
- La valeur d’En-tête Suivant qui identifie le premier en-tête de la partie fragmentable du paquet original.
- Un offset de fragmentation contenant l’offset du fragment, en nombre de mots de 8 octets, par rapport au début de la partie fragmentable du paquet original. L’offset de fragmentation du premier fragment (“ le plus à gauche ”) est 0.
- Une valeur du drapeau P à 0 si le fragment est le dernier (“ le plus à droite ”), sinon une valeur du drapeau P à 1.
- La valeur d’identification générée pour le paquet original.

(3) Le fragment lui-même.

Les longueurs des fragments doivent être choisies telles que les paquets fragmentés résultants tiennent dans le MTU de chemin vers la (les) destination(s) du paquet.

Arrivés à destination, les paquets fragmentés sont rassemblés pour obtenir leur forme originale, non fragmentée, comme illustré :

Paquet original rassemblé :



Les règles suivantes gouvernent l'assemblage :

Un paquet original est rassemblé uniquement à partir des paquets fragmentés ayant les mêmes Adresse Source, Adresse Destination et Identification de Fragment.

La partie non fragmentable du paquet assemblé se compose de tous les en-têtes jusqu'à, mais sans l'inclure, l'en-tête de fragmentation du premier paquet fragmenté (il s'agit du paquet dont l'Offset de Fragmentation est zéro), avec les deux changements suivants :

- Le champ En-tête Suivant du dernier en-tête de la partie non fragmentable est obtenu à partir du champ En-tête Suivant de l'en-tête de fragmentation du premier fragment.
- La Longueur de la Charge Utile du paquet assemblé est calculée à partir de la longueur de la partie non fragmentable et de la longueur et de l'offset du dernier fragment. Par exemple, une formule pour calculer la Longueur de la Charge Utile du paquet original assemblé est :

  CU.orig = CU.prem – LF.prem – 8 + (8 * OF.der) + LF.der

  où

  CU.orig = champ Longueur de la Charge Utile du paquet assemblé.
  CU.prem = champ Longueur de la Charge Utile du premier paquet fragmenté.
  LF.prem = longueur du fragment suivant l'en-tête de fragmentation du premier paquet fragmenté.
  OF.der = champ Offset de Fragmentation de l'en-tête de fragmentation du dernier paquet fragmenté.
  LF.der = longueur du fragment suivant l'en-tête de fragmentation du dernier paquet fragmenté.

La partie fragmentable du paquet assemblé est construite à partir des fragments suivants les en-têtes de fragmentation de chacun des paquets fragmentés. La longueur de chaque fragment est calculée en soustrayant à la Longueur de la 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 la valeur de l'offset de fragmentation.

L'en-tête de fragmentation n'est pas présent dans le paquet final, assemblé.

Les conditions d'erreur suivantes peuvent survenir lors de l'assemblage des paquets fragmentés :

Si un nombre insuffisant de fragments a été reçu pour achever l'assemblage d'un paquet dans les 60 secondes qui suivent l'arrivée du premier fragment de ce paquet, l'assemblage de ce paquet doit être abandonné et tous les fragments de ce paquet qui ont été reçus doivent être éliminés. Si le premier fragment (i.e., celui avec un Offset de Fragmentation à zéro) a été reçu, un message ICMP Délai Dépassé – Délai d'Assemblage des Fragments Dépassé (Time exceeded – Fragment Reassembly Time Exceeded) devrait être envoyé à la source de ce fragment.

Si la longueur d'un fragment, étant dérivée du champ Longueur de la Charge Utile du paquet fragmenté, n'est pas un multiple de 8 octets et que

le drapeau P de ce fragment est 1, alors ce fragment doit être éliminé et un message ICMP Problème de Paramètre, Code 0, pointant sur le champ Longueur de la Charge Utile du paquet fragmenté devrait être envoyé à la source du fragment.

Si la longueur et l'offset d'un fragment sont tels que la Longueur de la Charge Utile du paquet assemblé à partir de ce fragment dépasserait 65 535 octets, alors ce fragment doit être éliminé et un message ICMP Problème de Paramètre, Code 0, pointant sur le champ Offset de Fragmentation du paquet fragmenté devrait être envoyé à la source du fragment.

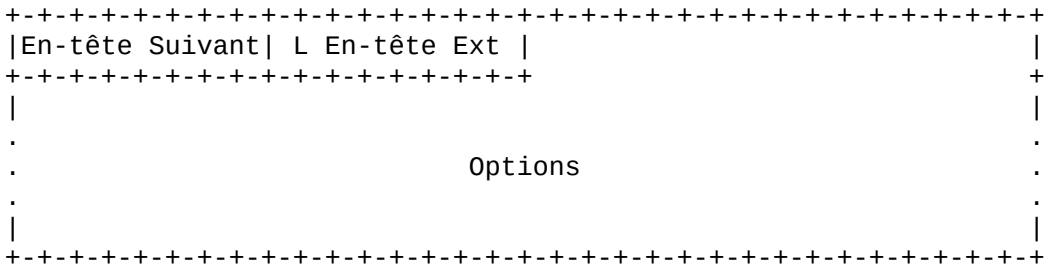
Les conditions suivantes ne sont pas sensées apparaître, mais elles ne sont pas considérées comme des erreurs si elles se produisent :

Le nombre et le contenu des en-têtes précédant l'en-tête de fragmentation de différents fragments du même paquet original peuvent différer. Quels que soient les en-têtes qui sont présents, précédant l'en-tête de fragmentation de chaque paquet fragmenté, ces en-têtes sont traités quand les paquets arrivent, avant de ranger ces paquets dans une file d'assemblage. Seuls les en-têtes du paquet fragmenté ayant l'Offset à zéro sont conservés dans le paquet assemblé.

Les valeurs d'En-tête Suivant des en-têtes de fragmentation des différents fragments du même paquet original peuvent différer. Seule celle du paquet fragmenté ayant l'Offset à zéro est utilisée pour l'assemblage.

4.6. En-tête des options de destination

L'en-tête des options de destination est utilisé pour transporter des informations optionnelles qui ont besoin d'être examinées uniquement par le(s) nœud(s) destination du paquet. L'en-tête des options de destination est identifié par une valeur d'En-tête Suivant à 60 dans l'en-tête le précédent immédiatement. Il a le format suivant :



En-tête Suivant (Next Header)

Sélecteur sur 8 bits. Identifie le type de l'en-tête suivant immédiatement l'en-tête des options de destination. Utilise les mêmes valeurs que le champ " protocole " d'IPv4 [RFC-1700 et suivants].

L En-tête Ext (Hdr Ext Len)

Entier 8 bits non signé. Longueur de l'en-tête des options de destination en mots de 8 octets, sans compter les 8 premiers octets.

Options (Options)

Champ de longueur variable, telle que l'en-tête des options de destination complet soit un entier multiple de 8 octets. Contient au moins une option encodée TLV, comme décrit dans la section 4.2.

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

Il faut noter qu'il y a deux façons possibles d'encoder des informations de destination optionnelles dans un paquet IPv6 : soit comme une option dans l'en-tête des options de destination, soit comme un en-tête d'extension séparé. L'en-tête de fragmentation et l'en-tête d'authentification sont des exemples de la dernière approche. Quelle approche peut être utilisée dépend de quelle action est désirée de la part d'un nœud destination qui ne comprend pas l'information optionnelle :

- Si l'action désirée est que le nœud destination élimine le paquet et, uniquement si l'Adresse Destination du paquet n'est pas une adresse multicast, envoie un message ICMP Type Non reconnu (Unrecognized Type) à l'Adresse Source du paquet, alors l'information peut être encodée soit comme un en-tête séparé, soit comme une option dans l'en-tête des options de destination dont le Type d'Option a la valeur 11 dans ses deux bits de poids fort. Le choix peut se fonder sur des facteurs tels que l'occupation en octets la plus faible, le meilleur alignement ou l'analyse la plus optimisée possible.

- Si une toute autre action est désirée, l'information doit être encodée comme une option dans l'en-tête des options de destination, avec un valeur de Type d'Option à 00, 01 ou 10 dans les deux bits de poids fort, spécifiant l'action désirée (voir section 4.2).

4.7. Pas d'en-tête suivant

La valeur 59 dans le champ En-tête Suivant d'un en-tête IPv6 ou tout autre en-tête d'extension indique que rien ne suit cet en-tête. Si le champ Longueur de la Charge Utile de l'en-tête IPv6 indique la présence d'octets après la fin de l'en-tête dont le champ En-tête Suivant contient 59, ces octets doivent être ignorés et envoyés tels quels si le paquet doit être transmis.

5. Problème de la longueur des paquets

IPv6 exige que chaque lien inter-réseaux ait un MTU supérieur ou égal à 1280 octets. Sur tout lien qui ne peut pas transporter un paquet de 1280 octet en un seul morceau, les services de fragmentation et d'assemblage spécifique au lien doivent être fournis par la couche en-dessous d'IPv6.

Les liens qui ont un MTU configurable (par exemple, les liens PPP [RFC-1661]) doivent être configurés pour avoir un MTU d'au moins 1280 octets ; il est recommandé qu'ils soient configurés avec un MTU supérieur ou égal à 1500 octets, pour s'accommoder avec d'éventuelles encapsulations (i.e., des tunnels) sans recourir à la fragmentation de la couche IPv6.

Un nœud doit être capable d'accepter des paquets aussi larges que le MTU du lien par lequel le paquet arrive et ceci pour chacun des liens auquel le nœud est directement attaché.

Il est fortement recommandé que les nœuds IPv6 implémentent la découverte du MTU de chemin [RFC-1981], dans le but de découvrir et de bénéficier d'un MTU de chemin supérieur à 1280 octets. Cependant, une implémentation minimale d'IPv6 (par exemple, dans une ROM de boot) peut simplement se restreindre lui-même à envoyer des paquets pas plus grand que 1280 octets et omettre l'implémentation de la découverte du MTU de chemin.

Afin d'envoyer un paquet plus large qu'un MTU de chemin, un nœud peut utiliser l'en-tête de fragmentation d'IPv6 pour fragmenter le paquet au niveau de la source ; le paquet sera rassemblé au niveau de la (des) destination(s). Cependant, l'utilisation d'une telle fragmentation n'est pas encouragée pour toute application qui est capable d'ajuster la taille de ses paquets pour tenir dans le MTU de chemin mesuré (i.e., jusqu'à 1280 octets).

Un nœud doit être capable d'accepter un paquet fragmenté qui, après assemblage, a une taille jusqu'à 1500 octets. Un nœud est autorisé à accepter des paquets fragmentés dont la taille une fois assemblés est supérieure à 1500 octets. Un protocole de couche supérieure ou une application qui compte sur la fragmentation d'IPv6 pour envoyer des paquets plus larges que le MTU d'un chemin ne devrait pas envoyer de paquets plus larges que 1500 octets sans avoir l'assurance que la destination est capable de rassembler des paquets de cette taille.

En réponse à un paquet IPv6 qui est envoyé à une destination IPv4 (i.e., un paquet qui subit une transcription d'IPv6 à IPv4), le nœud IPv6 d'origine peut recevoir un message ICMP Paquet Trop Gros (Packet Too Big), reportant un MTU de saut suivant plus petit que 1280 octets. Dans ce cas, le nœud IPv6 n'est pas obligé de réduire la taille des paquets significatifs à moins de 1280 octets, mais doit inclure un en-tête de fragmentation dans ces paquets pour que le routeur de transcription d'IPv6 à IPv4 puisse obtenir une valeur d'Identification valide à utiliser dans les fragments IPv4 résultants. Il faut noter que cela signifie que la charge utile peut devoir être réduite à 1232 octets (1280 moins 40 d'en-tête IPv6 moins 8 d'en-tête de fragmentation), ou réduite plus encore suivant le nombre d'en-têtes d'extension additionnels utilisés.

6. Labels relatifs aux flux d'information

Le champ Label du Flux sur 20 bits dans l'en-tête IPv6 peut être utilisé par une source pour nommer des séquences de paquets pour lesquels un traitement spécial de la part des routeurs IPv6 est demandé. Ce traitement spécial pourrait être une qualité de service différente du service par défaut ou un service " temps réel ". Cet aspect d'IPv6 est encore expérimental, au moment où ces lignes sont écrites, et sujet à des changements lorsque les exigences de support de flux sur Internet s'éclairciront. Les hôtes ou les routeurs qui ne supportent pas les fonctions du champ Label de Flux ont l'exigence de fixer le champ à zéro quand ils sont à l'origine du paquet, de passer le champ tel quel quand ils transmettent un paquet et d'ignorer le champ quand ils reçoivent un paquet.

L'Appendice A décrit la sémantique et l'utilisation du champ Label du Flux actuellement projetés.

7. Classes de trafic

Le champ Classe du Trafic sur 8 bits dans l'en-tête IPv6 a été créé pour être utilisé par des nœuds origines et/ou des routeurs transmetteurs pour identifier et distinguer différentes classes ou priorités de paquets IPv6. Au moment où cette spécification est écrite, il y a de nombreuses expérimentations en cours d'utilisation du Type de Service (Type of Service) et/ou des bits de Priorité (Precedence bits) pour fournir des formules diverses de " services différenciés " pour des paquets IP, autres qu'à travers l'utilisation de configurations explicites de flux. Le champ Classe du Trafic dans l'en-tête IPv6 est proposé pour que des fonctionnalités similaires puissent être supportées par IPv6. Il est espéré que ces expériences vont par la suite aboutir à une entente sur les sortes de classifications de trafic qui sont les plus utiles pour des paquets IPv6. Les définitions détaillées de la syntaxe et de la sémantique de tout ou partie des bits des Classes de Trafic d'IPv6, qu'ils soient expérimentaux ou dans une phase de standardisation, sont fournies dans des documents séparés.

Les exigences générales suivantes s'appliquent au champ Classe du Trafic :

- L'interface de service du service IPv6 d'un nœud doit proposer un moyen pour un protocole de couche supérieure de passer la valeur des bits de Classe du Trafic pour les paquets issus de ce protocole de couche supérieure. La valeur par défaut doit être zéro pour tous les 8 bits.

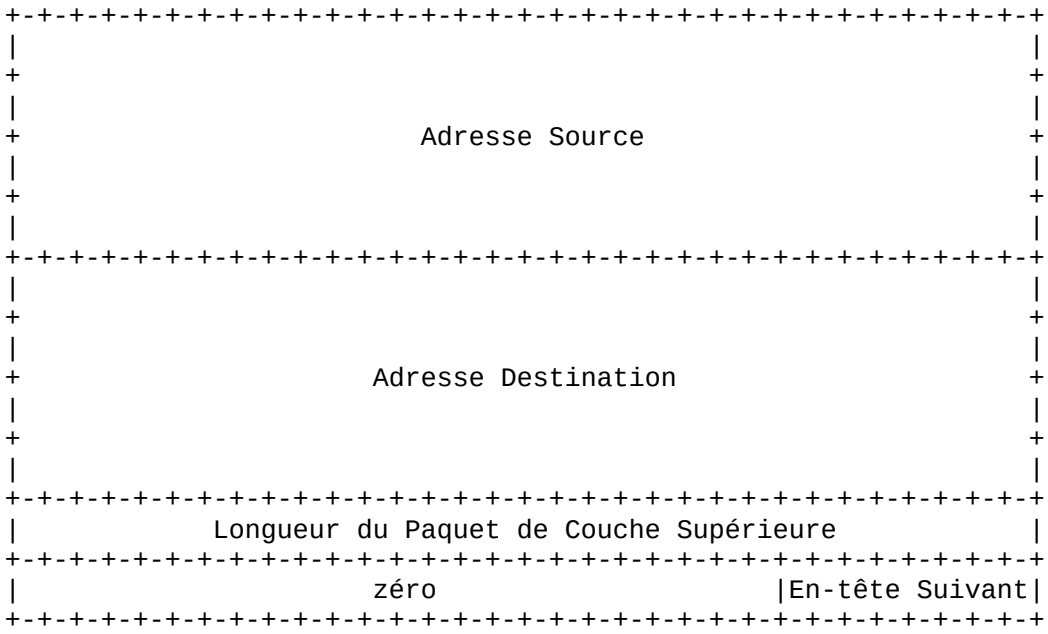
- Les nœuds qui supportent une utilisation spécifique (expérimentale ou éventuellement standard) de tout ou partie des bits de Classe du Trafic sont autorisés à changer la valeur de ces bits dans les paquets qu'ils créent, transmettent ou reçoivent, comme l'exige cette utilisation spécifique. Les nœuds devraient ignorer et laisser tel quel tout bit du champ Classe du Trafic pour lequel ils ne supportent pas une utilisation spécifique.

- Un protocole de couche supérieure ne doit pas supposer que la valeur des bits de la Classe du Trafic dans un paquet reçu est la même que la valeur envoyée par la source du paquet.

8. Problèmes relatifs aux protocoles de la couche supérieure

8.1. Sommes de contrôle de la couche supérieure

Tout protocole de transport ou d'une autre couche supérieure qui inclut les adresses IP (obtenues à partir de l'en-tête IP) dans le calcul de sa somme de contrôle doit être modifié pour être utilisé au-dessus d'IPv6, pour inclure les adresses IPv6 sur 128 bits à la place des adresses IPv4 sur 32 bits. En particulier, l'illustration suivante présente le " pseudo-en-tête " (" pseudo-header ") de TCP et UDP pour IPv6 :



- Si le paquet IPv6 contient un en-tête de routage, l'Adresse Destination utilisée dans le pseudo-en-tête est celle de la destination finale. Au niveau du nœud origine, cette adresse sera dans le dernier élément de l'en-tête de routage ; au niveau de(s) destinataire(s), cette adresse sera dans le champ Adresse Destination de l'en-tête IPv6.

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

- La Longueur du Paquet de Couche Supérieure dans le pseudo-en-tête est la longueur de l'en-tête de couche supérieure plus celle des données (par exemple, l'en-tête TCP plus les données TCP). Certains protocoles de couche supérieure transportent leur propre information longueur (par exemple, le champ Longueur dans l'en-tête UDP) ; pour de tels protocoles, c'est la longueur qui est utilisée dans le pseudo-

en-tête. D'autres protocoles (comme TCP) ne transportent pas leur propre information longueur, dans ce cas la longueur utilisée dans le pseudo-en-tête est la Longueur de la Charge Utile de l'en-tête IPv6, moins la longueur de tous les en-têtes d'extension présents entre l'en-tête IPv6 et l'en-tête de couche supérieure.

- Contrairement à IPv4, quand des paquets UDP sont issus d'un nœud IPv6, la somme de contrôle UDP n'est pas optionnelle. Pour cela, chaque fois qu'un paquet UDP est créé, le nœud IPv6 doit calculer la somme de contrôle UDP sur le paquet et le pseudo-en-tête et si le résultat est zéro, celui-ci doit être changé en la valeur hexadécimale FFFF avant de le placer dans l'en-tête UDP. Les destinataires IPv6 doivent éliminer les paquets UDP contenant une somme de contrôle à zéro et devraient reporter l'erreur.

La version IPv6 de ICMP [ICMPv6] inclut le pseudo-en-tête précédent dans son calcul de la somme de contrôle ; c'est un changement par rapport à la version IPv4 de ICMP, qui n'incluait pas un pseudo-en-tête dans sa somme de contrôle. La raison de ce changement est de protéger ICMP d'erreurs d'acheminement ou de corruptions des champs de l'en-tête IPv6 desquels il dépend, lesquels, contrairement à IPv4, ne sont pas protégés par une somme de contrôle de la couche inter-réseaux. Le champ En-tête Suivant dans le pseudo-en-tête pour ICMP contient la valeur 58, qui identifie la version IPv6 d'ICMP.

8.2. Durée de vie maximale d'un paquet

Contrairement à IPv4, les nœuds IPv6 ne sont pas obligés d'imposer un temps de vie maximum des paquets. C'est la raison pour laquelle le champ IPv4 " Temps à Vivre " (" Time to Live ") a été renommé " Nombre de Sauts Maximum " (" Hop Limit ") dans IPv6. En pratique, très peu, s'il y en a, d'implémentations d'IPv4 sont conformes aux exigences de limitation de la durée de vie des paquets, aussi ceci n'est pas un changement en pratique. Tout protocole de couche supérieure qui compte sur la couche inter-réseaux (que se soit IPv4 ou IPv6) pour limiter la durée de vie des paquets devrait être mis à jour pour fournir ses propres mécanismes pour détecter et éliminer des paquets périmés.

8.3. Taille maximale de la charge utile de la couche supérieure

Lors du calcul de la taille maximale disponible pour la charge utile des données de la couche supérieure, un protocole de couche supérieure doit prendre en compte la taille plus importante de l'en-tête IPv6 par rapport à l'en-tête IPv4. Par exemple, avec IPv4, l'option MSS de TCP est calculée comme la taille maximale du paquet (une valeur par défaut ou une valeur déduite de la découverte du MTU de chemin) moins 40 octets (20 octets pour la longueur minimale de l'en-tête IPv4 plus 20 octets pour la longueur minimale de l'en-tête TCP). Lorsque TCP est utilisé au-dessus d'IPv6, le MSS doit être calculé comme la taille maximale du paquet moins 60 octets car la longueur minimale de l'en-tête IPv6 (i.e., un en-tête IPv6 sans en-têtes d'extension) est 20 octets plus grande que la longueur minimale de l'en-tête IPv4.

8.4. Comment répondre aux paquets transportant des en-têtes de routage

Quand un protocole de couche supérieure envoie un ou plusieurs paquets en réponse à un paquet reçu qui inclus un en-tête de routage, le(s) paquet(s) de réponse ne doivent pas inclure un en-tête de routage obtenu automatiquement par une " inversion " (" reversing ") de l'en-tête de routage reçu A MOINS QUE l'intégrité et l'authenticité de l'Adresse Source et de l'en-tête de routage reçus aient été vérifiées (par exemple, en utilisant dans le paquet reçu un en-tête d'authentification). En d'autres termes, seuls les types de paquets suivants sont permis en réponse à un paquet reçu contenant un en-tête de routage :

- des paquets de réponse qui ne contiennent pas d'en-têtes de routage.

- des paquets de réponse qui contiennent des en-têtes de routage NON dérivés d'une inversion de l'en-tête de routage du paquet reçu (par exemple, un en-tête de routage fourni par une configuration locale).

- des paquets de réponse qui contiennent des en-têtes de routage qui sont dérivés d'une inversion d'en-tête de routage d'un paquet reçu SI ET UNIQUEMENT SI l'intégrité et l'authenticité de l'Adresse Source et de l'en-tête de routage du paquet reçu ont été vérifiées par celui qui répond.

Appendice A. Sémantique et utilisation du champ "label du flux"

Un flux est une séquence de paquets envoyée par une source particulière vers une destination particulière (unicast ou multicast) pour lequel la source désire un traitement spécial de la part des routeurs intermédiaires. La nature de ce traitement spécial pourrait être communiqué aux routeurs par un protocole de contrôle, tel qu'un protocole de réservation de ressources, ou par des informations à l'intérieur même des paquets du flux, par exemple, dans une option sauts après sauts. Les détails de tels protocoles de contrôle ou de telles options dépassent le cadre de ce document.

Il peut y avoir de multiples flux actifs d'une source vers une destination, aussi bien que du trafic non associé à un flux particulier. Un flux est identifié de manière unique par la combinaison d'une adresse source et d'un label de flux différent de zéro. Des paquets qui ne font pas partie d'un flux particulier ont un label de flux à zéro.

Un label de flux est assigné à un flux par le nœud source du flux. Les nouveaux labels de flux doivent être choisis (pseudo-)aléatoirement et uniformément répartis entre 1 et FFFFF hexadécimal. Le but de cette allocation aléatoire est de fabriquer n'importe quel ensemble de bits à l'intérieur du champ Label du Flux apte à être utilisé comme une clé pour une table de hachage (hash key) par les routeurs, pour remonter à l'état associé au flux.

Tous les paquets appartenant à un même flux doivent être envoyés avec les mêmes adresse source, adresse destination et label de flux. Si l'un quelconque de ces paquets inclut un en-tête des options sauts après sauts, alors ils doivent tous être créés avec exactement le même en-tête des options sauts après sauts (mis à part le champ En-tête Suivant). Si l'un quelconque de ces paquets inclut un en-tête de routage, alors ils doivent tous être créés avec exactement les mêmes en-têtes d'extension, jusqu'à et incluant l'en-tête de routage (mis à part le champ En-tête Suivant de l'en-tête de routage). Les routeurs ou les destinations ont la possibilité, mais ne sont pas obligés, de vérifier que ces conditions sont satisfaites. Si une violation est détectée, elle devrait être reportée à la source par un message ICMP Problème de Paramètre, Code 0, pointant sur l'octet de poids fort du champ Label du Flux (i.e., un décalage de 1 à l'intérieur du paquet IPv6).

La durée de vie maximale d'un quelconque état de traitement de flux (flow-handling state) établi le long du chemin du flux, doit être spécifiée comme partie intégrante de la description du mécanisme d'établissement d'un état, par exemple, lors de la spécification d'un protocole de réservation de ressources ou d'une option sauts après sauts de configuration de flux. Une source ne doit pas réutiliser un label de flux pour un nouveau flux pendant la durée de vie maximale d'un quelconque état de traitement de flux qui aurait pu être établi antérieurement pour l'utilisation de ce label de flux.

Lorsqu'un nœud s'arrête et redémarre (par exemple, après un " plantage "), la précaution doit être prise de ne pas réutiliser un label de flux qui aurait pu être utilisé par un flux précédent dont la durée de vie peut ne pas avoir encore expiré. Ceci peut être accompli en enregistrant sur un support stable l'utilisation des labels de flux permettant au nœud de s'en souvenir même après un plantage, ou en s'abstenant d'utiliser tout label de flux jusqu'à ce que la durée de vie maximale d'un quelconque flux susceptible d'avoir été établi précédemment ait expiré. Si le temps minimum pour

redémarrer un nœud est connu, ce temps peut être déduit de la période d'attente nécessaire pour commencer à allouer des labels de flux.

Il n'y a aucune exigence sur le fait que tous les paquets (ou même la majorité des paquets) appartiennent à un flux, i.e., aient un label de flux différent de zéro. Cette observation est placée ici pour rappeler aux personnes mettant en œuvre ou concevant des protocoles de ne pas supposer le contraire. Par exemple, il serait peu judicieux de concevoir un routeur dont les performances seraient adéquates uniquement si la plupart des paquets appartiennent à des flux ou de concevoir une compression d'en-têtes qui travaille uniquement sur des paquets qui appartiennent à des flux.

Appendice B. Instructions pour le formatage des options

Cette appendice donne quelques conseils sur comment arranger les champs lors de la conception de nouvelles options pouvant être utilisées dans l'en-tête des options sauts après sauts ou dans l'en-tête des options de destination, comme décrit dans la section 4.2. Ces instructions sont basées sur les hypothèses suivantes :

- Une caractéristique souhaitable est que tout champ de plusieurs octets à l'intérieur de l'espace des Données de l'Option d'une option soit aligné sur sa frontière naturelle, i.e., les champs de n octets de large devraient être placés à un multiple entier de n octets à partir du début de l'en-tête des options sauts après sauts ou de destination, pour n = 1, 2, 4 ou 8.

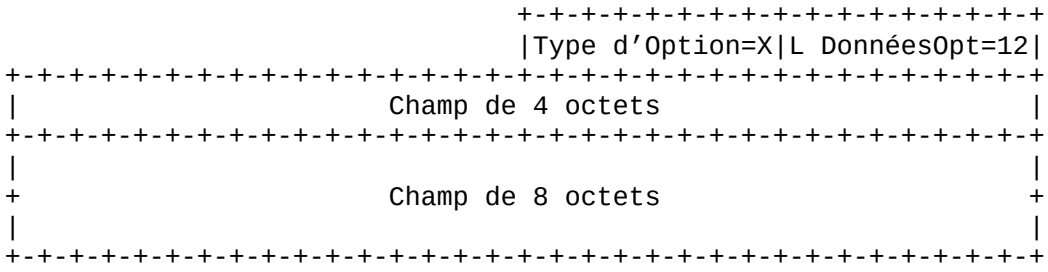
- Une autre caractéristique souhaitable est que l'en-tête des options de destination ou des options sauts après sauts prenne le moins d'espace possible (tout en respectant l'exigence que la longueur de l'en-tête est un multiple entier de 8 octets).

- Il peut être supposé que, lorsque l'un de ces deux en-têtes transportant des options est présent, il transporte un nombre très restreint d'options, en général une seule.

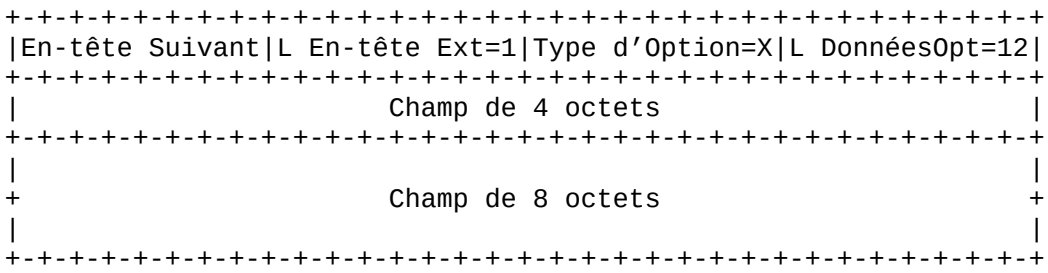
Ces hypothèses suggèrent l'approche suivante pour organiser les champs d'une option : ordonner les champs du plus petit au plus large, sans caractères de bourrage à l'intérieur ; puis, tirer des exigences d'alignement pour l'option entière basées sur l'exigence d'alignement du champ le plus large (jusqu'à un alignement maximum de 8 octets). Cette approche est illustrée dans les exemples suivants :

Exemple 1

Si une option X exige deux champs de données, un de 8 octets de longueur et un de 4, elle pourrait être organisée comme ci-après :

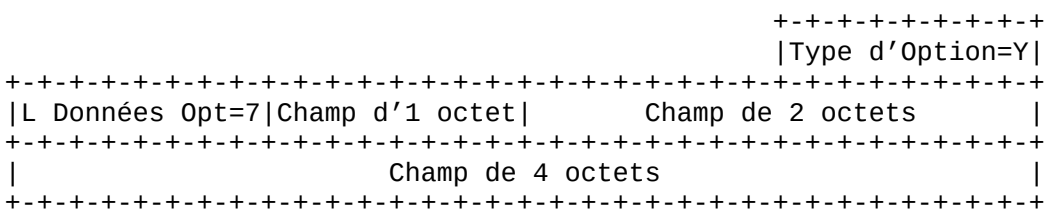


Son exigence d'alignement est 8n+2, pour assurer que le champ de 8 octets commence à un décalage multiple de 8 octets à partir du début de l'en-tête englobant. Un en-tête des options sauts après sauts ou des options de destination complet contenant cette seule option ressemblera à celui ci-après :

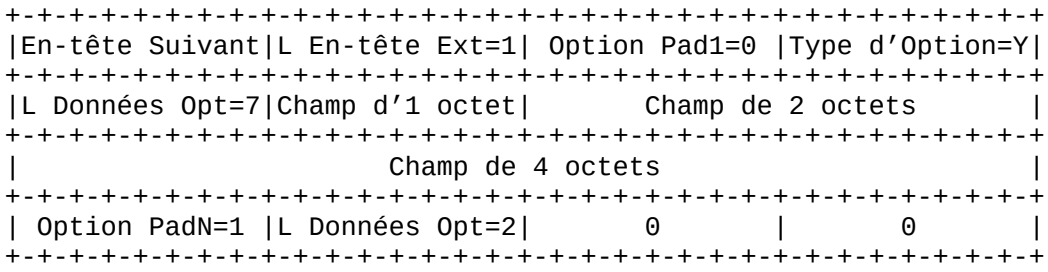


Exemple 2

Si une option Y exige trois champs de données, un d’une longueur de 4 octets, un de 2 et un de 1, elle pourrait être organisée comme ci-après :

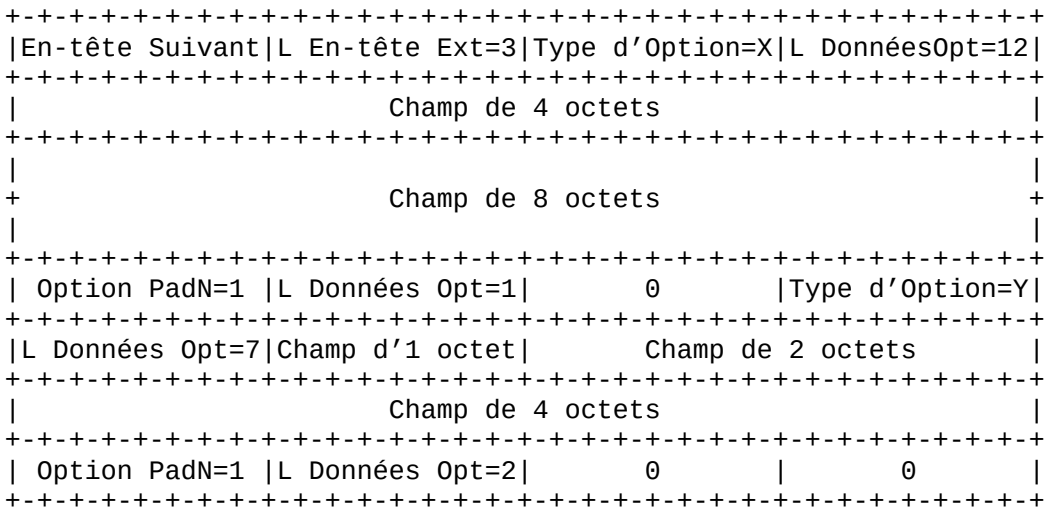


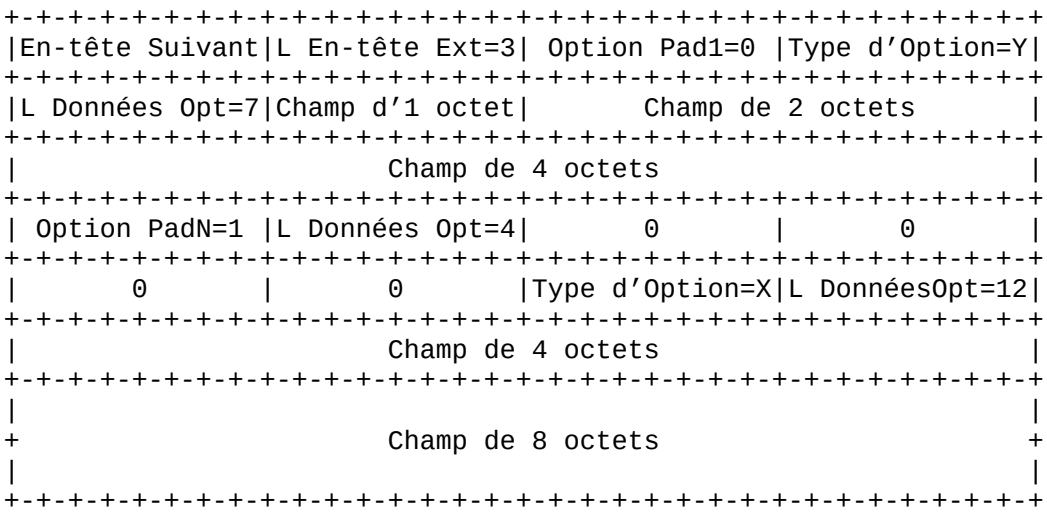
Son exigence d’alignement est 4n+3, pour assurer que le champ de 4 octets commence à un décalage multiple de 4 octets à partir du début de l’en-tête englobant. Un en-tête des options sauts après sauts ou des options de destination complet contenant cette seule option ressemblera à celui ci-après :



Exemple 3

Un en-tête des options sauts après sauts ou des options de destination contenant à la fois l’option X et Y des exemples 1 et 2 aurait un des deux formats suivants, en fonction de l’option qui apparaît en premier :





Considérations sur la sécurité

Les caractéristiques de sûreté de IPv6 sont décrites dans “ Considérations architecturales sur la sûreté du Protocole Internet ” [RFC-2401].

Remerciements

Les auteurs remercient avec reconnaissance les membres du groupe de travail IPng, le groupe de recherche sur les Protocoles Point-à-Point et la Communauté Internet au sens large pour ses nombreuses et utiles suggestions.

Adresses des Auteurs

Stephen E. Deering
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
Phone : +1 408 527 8213
Fax : +1 408 527 8254
E-Mail: deering@cisco.com

Robert M. Hinden
Nokia
232 Java Drive
Sunnyvale, CA 94089
USA
Phone : +1 408 990 2004
Fax : +1 408 743 5677
E-Mail: hinden@iprg.nokia.com

Références

[RFC-2401] S. Kent et R. Atkinson, “ Considérations architecturales sur la sûreté du Protocole Internet ” (“ Security Architecture for the Internet Protocol ”), RFC 2401, novembre 1998.

[RFC-2402] S. Kent et R. Atkinson, “ En-tête IP d’authentification ” (“ IP Authentication Header ”), RFC 2402, novembre 1998.

[RFC-2406] S. Kent et R. Atkinson, “ Protocole de sûreté à l’intérieur d’IP (ESP)” (“ IP Encapsulating Security Protocol (ESP) ”), RFC 2406, novembre 1998.

[ICMPv6] A. Conta et S. Deering, “ ICMP pour le Protocole Internet Version 6 (IPv6) ” (“ ICMP for the Internet Protocol Version 6 (IPv6) ”), RFC 2463, décembre 1998.

[ADDRARCH] R. Hinden et S. Deering, “ Architecture d’adressage sur IP Version 6 ” (“ IP Version 6 Addressing Architecture ”), RFC 2373, juillet 1998.

[RFC-1981] J. Mac Cann, J. Mogul et S. Deering, “ Découverte du MTU de chemin pour IP version 6 ” (“ Path MTU Discovery for IP version 6 ”), RFC 1981, août 1996.

[RFC-791] J. Postel, “ Le Protocole Internet ” (“ Internet Protocol ”), STD 5, RFC 791, septembre 1981.

[RFC-1700] J. Reynolds et J. Postel, “ Nombres Réservés ” (“ Assigned Numbers ”), STD 2, RFC 1700, octobre 1994. Voir aussi : http://www.iana.org/numbers.html

[RFC-1661] W. Simpson, “ Le Protocole Point-à-Point (PPP) ” (“ The Point-to-Point Protocol (PPP) ”), STD 51, RFC 1661, juillet 1994.

Différences par rapport au RFC-1883

Ce document contient les changements suivants par rapport au RFC-1883. Des nombres identifient la version du travail en cours sur lequel s’appliquent ces modifications.

02) Retrait de toute référence aux jumbogrammes (jumbograms) et à l’option de Charge Utile Jumbo (Jumbo Payload option) (déplacé dans un document séparé).

02) Déplacement de la majorité de la description du Label du Flux de la section 6 à la (nouvelle) Appendice A.

02) Dans la description du Label du Flux, à présent dans l’Appendice A, correction de la valeur maximale du Label du Flux de FFFFFF à FFFFF (i.e., un “ F ” de moins) due à la réduction de la taille du champ Label du Flux de 24 bits à 20 bits.

02) Renumérotation (Relettration ?) de l’ancien Appendice A en l’actuel Appendice B.

02) Changement de formulation de la section Considérations sur la sécurité pour éviter une boucle d’interdépendance entre cette spec et les specs d’IPsec.

02) Mise à jour de l’adresse électronique et de l’entreprise d’affiliation de R. Hinden.

---

01) Dans la section 3, changement du champ nommé “ Classe ” (“ Class ”) en “ Classe du Trafic ” et augmentation de sa taille, de 4 à 8 bits. Réduction de la taille du champ Label du Flux de 24 à 20 bits pour compenser l’augmentation du champ Classe du Trafic.

01) Dans la section 4.1, restauration de l'ordre de l'en-tête d'Authentification et de l'en-tête ESP, qui ont été intervertis par erreur dans la version 00 de ce document.

01) Dans la section 4.4, retrait du champ Champ de Bits Strict/Vague (Strict/Loose Bit Map) et de la fonctionnalité de routage strict de l'en-tête de routage Type 0, et retrait de la restriction sur le nombre d'adresses que peut transporter l'en-tête de routage de Type 0 (avait été limité à 23 adresses, à cause de la taille du champ de bits Strict/Vague).

01) Dans la section 5, changement du MTU IPv6 minimum de 576 à 1280 octets, et ajout d'une recommandation pour que les liens dont le MTU est configurable (par exemple, les liens PPP) soient configurés pour avoir un MTU d'au moins 1500 octets.

01) Dans la section 5, retrait de l'exigence concernant le fait qu'un nœud ne doit pas envoyer de paquets fragmentés dont l'assemblage nécessite plus de 1500 octets sans connaître la taille du buffer d'assemblage de la destination, remplacée par une recommandation sur le fait que les protocoles de couche supérieure ou les applications ne devraient pas faire cela.

01) Remplacement de la référence à la spec sur la Découverte du MTU de Chemin d'IPv4 (RFC-1191) par la référence à la spec sur la Découverte du MTU de Chemin d'IPv6 (RFC-1981), et retrait des Notes à la fin de la section 5 à propos de la Découverte du MTU de Chemin, depuis que ces détails sont abordés à présent par la RFC-1981.

01) Dans la section 6, retrait de la spécification de la configuration du flux " opportuniste " (" opportunistic " flow set-up), et retrait de toutes les références à la durée de vie maximale de 6 secondes pour un état du flux établi de façon opportuniste.

01) Dans la section 7, retrait de la description provisoire de la structure interne et de la sémantique du champ Classe du Trafic, et spécification du fait que de telles descriptions sont fournies dans des documents séparés.

---

00) Dans la section 4, correction de la valeur Code pour indiquer " Rencontre d'un type d'En-tête Suivant inconnu " avec un message ICMP Problème de Paramètre (changé de 2 à 1).

00) Dans la description du champ Longueur de la Charge Utile de la section 3, et celle du champ Longueur de la Charge Utile Jumbo de la section 4.3, reformulation plus claire du fait que les en-têtes d'extension font partis de la charge utile et sont comptabilisés en tant que telle.

00) Dans la section 4.1, interversion de l'ordre des en-têtes d'Authentification et ESP. (NOTE : c'était une erreur et ce changement a été annulé dans la version 01.)

00) Dans la section 4.2, reformulation plus claire du fait que les options sont identifiées par tous les 8 bits du Type d'Option, et non pas par les 5 bits de poids faible d'un Type d'Option. Egalement, spécification du fait que le même espace de numérotation de Type d'Option est utilisé pour les en-têtes des options sauts après sauts et des options de destination.

00) Dans la section 4.4, ajout d'une phrase exigeant que les nœuds traitant un en-tête de routage doivent envoyer un message ICMP Paquet Trop Gros en réponse à un paquet qui est trop gros pour passer par le lien sur lequel aurait eu lieu le prochain saut (plutôt que de dire de faire une fragmentation).

00) Changement du nom du champ Priorité (Priority) d'IPv6 en " Classe ", et remplacement de la description précédente de Priorité de la section 7 par une description du champ Classe. Egalement, retrait de ce champ de l'ensemble des champs qui doivent rester inchangés pour tous les paquets à l'intérieur d'un même flux, comme spécifié dans la section 6.

00) Dans le pseudo-en-tête dans la section 8.1, changement du nom du champ " Longueur de la Charge Utile " en " Longueur du Paquet de Couche Supérieure ". Clarification également du fait que, dans le cas des protocoles qui transportent leur propre information de longueur (comme le non-jumbogramme UDP), c'est la longueur dérivée de la couche supérieure, et non la longueur dérivée de la couche IP, qui est utilisée dans le pseudo-en-tête.

00) Ajout dans la section 8.4, pour spécifier que les protocoles de couche supérieure, lorsqu'ils répondent à un paquet reçu qui transporte un en-tête de routage, ne doivent pas inclure l'en-tête de routage inversé dans le(s) paquet(s) de réponse, à moins que l'en-tête de routage reçu ait été authentifié.

00) Correction de quelques erreurs typographiques et grammaticales.

00) Informations de contact des auteurs mises à jour.
