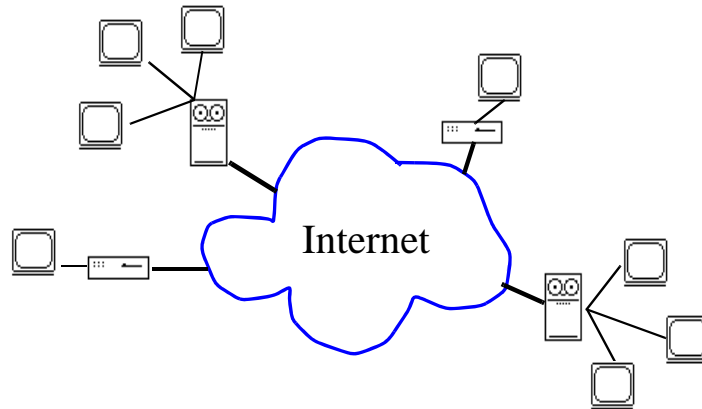


# Le protocole IP de nouvelle génération

(Z:\Polys\Internet\_transmission\_donnees\08-IPv6.fm- 3 octobre 2008 16:15)



## PLAN

1. Introduction
2. Le format général du paquet
  - 2.1 Les champs
  - 2.2 Les flots
3. Les adresses
4. Les entêtes optionnelles
  - 4.1 L'entête Hop-by-hop Options
  - 4.2 L'entête Routing
  - 4.3 L'entête Fragment
  - 4.4 L'entête Destination Options
5. La sécurité
6. Conclusion

## Bibliographie

- G. Cizault, "IPv6", Hermès, 2006.
- C. Huitema, "IPv6 : the new Internet Protocol", Prentice Hall, 1996.
- Scott & Allison, "IPng : Internet Protocol Next Generation", Addison-Wesley, 1995.

# 1. Introduction

## 1.1. Historique

“Internet protocol” - **IP v4** (rfc 781) : septembre 1981.

- . “Connectionless network protocol” - CLNP ou IP-ISO (OSI 8473):
    - homogénéisation des mondes OSI et Internet
  - . “TCP and UDP over Bigger Address” - TUBA (rfc 1347, 1526, 1561) :
    - IP basé sur CLNP
  - . “Simple IP” - SIP:
    - augmentation de l'espace d'adressage; technique de codage
  - . Pip (rfc 1621):
    - technique de routage : “source routing”, mobilité + adressage à longueur variable.
  - . “Internet IP Address Encapsulation” - IPAE :
    - stratégie de transition : IP -> IPng
  - . “Simple IP Plus” - SIPP (rfc 1710) :
    - convergence SIP + PIP
  - . “Common Architecture for the Internet” - CATNIP (Rfc 1707) :
    - CLNP + IP + IPX (Xerox)
- => “Internet protocol new generation” - **IP v6** (rfc 1883) : décembre 1995.

## 1.2. Principales modifications

### Extension des capacités d'adressage :

- . 32 bits >>128 bits
  - => + de niveaux hiérarchiques, + de noeuds
- . nouveau type d'adresse (cluster address)
  - => anycast (+ unicast + multicast)

### Simplification de l'entête :

- . de nombreux champs sont devenus optionnels
  - => diminue le surcoût (“overhead”) : temps de traitement et quantité de données

### Etiquetage des flots de données :

- . distinction entre les différents trafics
  - => améliore le routage

### Amélioration de la sécurité :

- . Authentification
- . Confidentialité des données

### Gestion de la mobilité

### 1.3. Terminologie

**Noeud :**

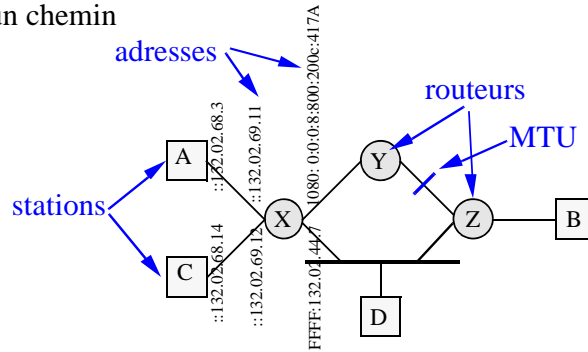
- routeurs : un noeud qui est capable de retransmettre les paquets qui ne lui sont pas destinés.
- stations : les autres noeuds.

**Liaison :** un moyen de communication utilisé par les noeuds pour communiquer

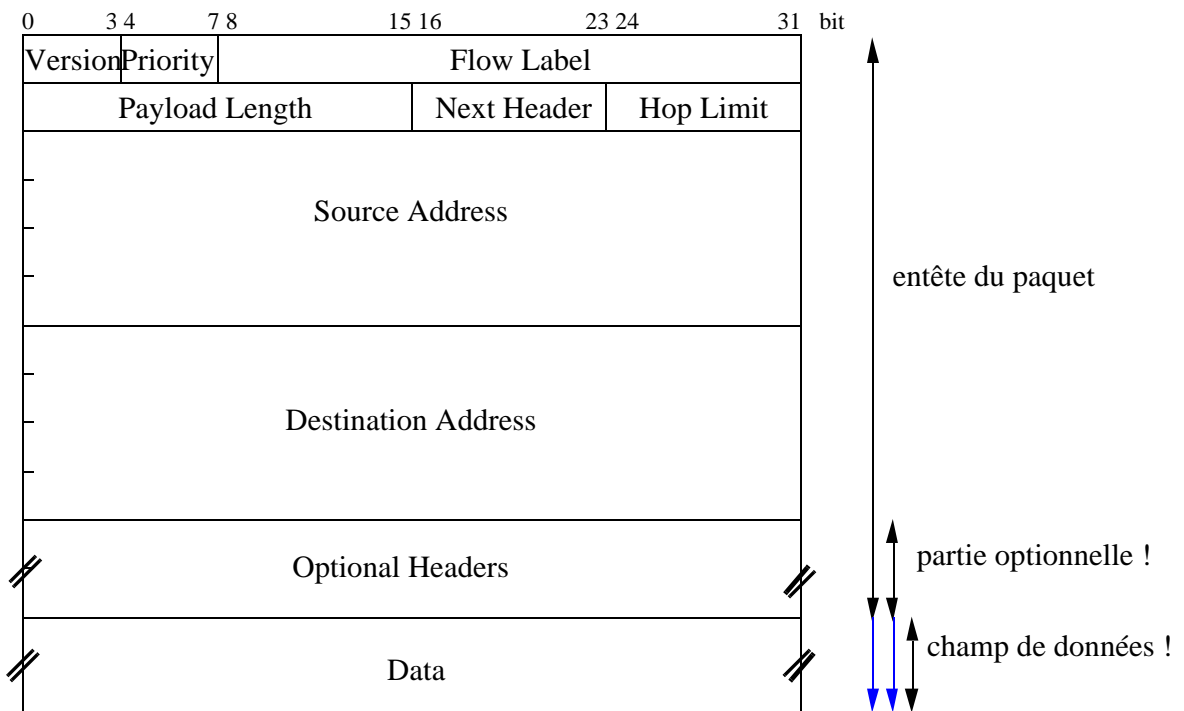
- interface : l'attachement d'un noeud à une liaison.
- adresse : l'identification d'une interface ou d'un ensemble d'interfaces.

**MTU (Maximum transmission unit) :**

- MTU d'une liaison
- MTU d'un chemin



## 2. Format général des paquets



## 2.1. Le champ Version

Le champ **Version** (4 bits) :

- Compatible avec IP v4 ! assure la transition :
  - . Les stations ou routeurs recevant un datagramme d'une version qu'ils ne décodent pas doivent l'écarter.
  - . Une même station ou routeur peut être capable de traiter plusieurs versions ! (“dual stack station”)
- Le code :
  - . 4 : “Internet protocol” - **IP v4** (rfc 781) : septembre 1981.
  - . 5 : “Stream protocol” - ST-II, ST2+ (rfc 1190, rfc 1819)
  - . 6 : “Internet protocol new generation” - **IP v6** (rfc 1883) : décembre 1995.

## 2.2. Le champ Priority

Le champ **Priority** (4 bits) :

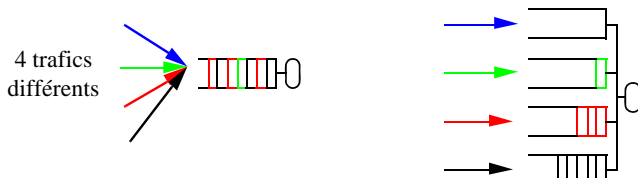
- . identifie les différents types de trafics.
- . similaire au champ “precedence” d'IP v4.
- . 2 classes de trafic en fonction de leur réaction vis-à-vis des congestions :
  - autorisant la retransmission des données (par ex. TCP)
  - la retransmission des données est inefficace (par ex. temps réel)
- . La classe des **trafics qui autorisent** le contrôle de la congestion :
  - 0 - trafic non caractérisé
  - 1 - trafic de remplissage “filler” (par ex. netnews).
  - 2 - transfert de données non attendues (par ex. email).
  - 3 - réservé
  - 4 - transfert massif de données attendues (par ex. FTP, NFS).
  - 5 - réservé
  - 6 - trafic interactif (par ex. Telnet, X)
  - 7 - trafic de contrôle du réseau Internet (par ex. protocoles de routage, SNMP)
- . La classe des **trafics qui ne permettent** pas le contrôle de la congestion :
  - 8 - priorité la plus basse (ces paquets seront les premiers éliminés si une congestion apparaît, par ex. trafic de haute fidélité vidéo).
  - 15 - priorité la plus haute (ces paquets seront les derniers éliminés, par ex. trafic de basse fidélité audio).

## 2.2.1 La gestion de la priorité

## “Priority queuing”

Ces différents types de trafics doivent être traités différemment dans les routeurs :

- Vis-à-vis des congestions
- Vis-à-vis des délais (vitesse de traitement)
- Similaire aux techniques de gestions des processus/processeur
- C'est optionnel !



Le taux de service associé à chaque type de trafic est :

- . [1] soit équivalent/indépendant de son type : FIFO
- . [2] soit déterminé par sa priorité

## 2.3. Les champs “Payload length” et “Hop limit”

Le champ **Payload Length** (2 octets) :

- longueur en octets de la charge utile du paquet (ce qui suit la partie fixe de l'entête IPv6).
  - < 64 Koctets !
  - 0 : indique que la longueur doit être trouvée dans l'option Jumbo Payload de l'entête Hop-by-hop.
- => similaire à l'option “Total length” d'IPv4.

Le champ **Hop Limit** (1 octet) :

- durée limite de vie du paquet :
    - . les paquets fantômes qui errent sans fin : erreur de routage
    - . limitation (grossière) du domaine atteignable par un paquet
  - décrémente à chaque routeur.
  - si la valeur atteint 0, le paquet est détruit.
- => similaire au champ “Time To Live” d'IPv4 (dont l'unité était la seconde).

## 2.4. Le champ “next Header”

Le champ **Next Header** (1 octet) :

- . Identifie le type de la **prochaine** entête optionnelle.
- . Chaînage des entêtes optionnelles.
- . Le champ de données est considéré comme une entête optionnelle !
- . Optimisation de la taille de l'entête et du temps de traitement :
  - => les options non utilisées ne sont pas présentes.

Equivalent au champ “protocol” de IPv4

## 2.5. Le champ “Flow label”

Le champ **Flow Label** (3 octets) :

Un flot == l'ensemble des paquets équi-étiquetés en provenance d'une même source.

- . L'utilisation du champ “Flow label” peut être d'utilisation plus souple que le champ “Priority”

Mais :

- . Dépendant de l'implémentation dans les routeurs.
- . Dépendant de l'utilisation de protocoles supplémentaires (RSVP) ou d'options d'IP v6 pour établir les flots.
- . L'interprétation du champ Flow label est optionnelle.

=> Flow label + @IP d'émission == 1 flot

### Le champ Flow label (suite)

- Tous les paquets d'un même flot doivent subir les mêmes traitements au sein des routeurs :
  - . les paramètres des paquets d'un même flot doivent être identiques (adresses, priority, options de l'entête Hop-by-hop, champs de l'entête Routing)
- Ces traitements peuvent être optimisés en utilisant le Flow Label et l'adresse source comme une clef d'accès à un descripteur :
  - . accès rapide, pré-traitement, etc.
- La nature du traitement spécifique doit être transmise aux routeurs à l'aide :
  - . d'un protocole spécifique de réservation de ressources (par ex. RSVP).
  - . des paquets eux-mêmes (par ex. à l'aide des options de l'entête Hop-by-hop)
- Un paquet ayant un champ Flow Label=0 n'est associé à aucun flot.
- L'optimisation des traitements donc l'utilisation du Flow Label est laissé libre (le paquet est alors considéré comme n'appartenant à aucun flot).
- Pour maintenir un flot en vie, un paquet de ce flot doit être périodiquement reçu (Toutes les 6 secondes au moins).

## 3. L'adressage IP v6

### 3.1. Les champs d'adresse d'IPv6

Les champs **Source and Destination Address** (16 octets) :

- La longueur du champ d'adresse passe de 32 bits à 128 !
- Doit faire face à un double accroissement (prospectif) :
  - . accroissement de nombre d'ordinateurs personnels
    - => de la population mondiale
    - => du taux d'utilisation
  - . accroissement du taux de présence dans les équipements
    - => les équipements seront contrôlables à distance
- Trois types d'adresse :
  - . adresse **unicast** : identifie une interface unique.
  - . adresse **anycast** : identifie le plus proche interface parmi un ensemble d'interfaces.
  - . adresse **multicast** : identifie un ensemble d'interfaces.

### 3.2. Représentation des adresses

- . Forme conventionnelle :  
 FEDC:BA98:7654:3210:FEDC:BA98:7654:3210  
 1080:0:0:0:8:800:200c:417A
- . Forme compressée :  
 1080:0:0:0:8:800:200c:417A --> 1080::8:800:200c:417A  
 FF01:0:0:0:0:0:43 --> FF01::43
- . Forme compatible avec IPv4 :  
 0:0:0:0:0:0:13.1.68.3  
 0:0:0:0:0:FFFF:129.144.52.38  
 permettra une transition douce : IPv4 => IPv6

### 3.3. Les différents modes d'attribution des adresses

- => le préfixe d'une adresse :  
 - les premiers bits d'une adresse définissent son attribution.

Attribution	Préfixe	Occupation
réservé	0000 0000	1/256
non-attribué	0000 0001	1/256
réservé pour NSAP	0000 001	1/128
réservé pour IPX	0000 010	1/128
non-attribué	0000 011	1/128
non-attribué	0000 1	1/32
non-attribué	0001	1/16
non-attribué	001	1/8
adresses unicast attribuées par fournisseurs	010	1/8
non-attribué	011	1/8
adresses unicast attribués géographiquement	100	1/8
non-attribué	101	1/8
non-attribué	110	1/8
non-attribué	1110	1/16
non-attribué	1111 0	1/32
non-attribué	1111 10	1/64
non-attribué	1111 110	1/128
non-attribué	1111 1110	1/256
non-attribué	1111 1110 0	1/512
adresses unicast attribuées au niveau d'une liaison	1111 1110 10	1/1024
adresses unicast attribuées au niveau d'un site	1111 1110 11	1/1024
adresses multicast	1111 1111	1/256

> 85%  
non-attribué  
ou réservé

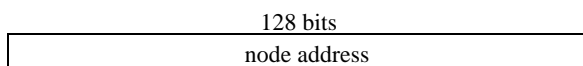


### 3.4. Adresses unicasts

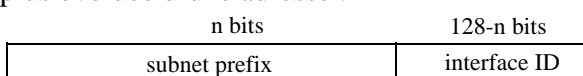
#### 3.4.1 Introduction

- . La structure interne des adresses est hiérarchique.
- . Les adresses d'un même domaine d'adressage sont
  - contiguës : construction de masques.
  - CIDR : Classless Interdomain Routing - similaire à IPv4 (rfc 1338).
- . La connaissance de cette structure dépend du rôle du noeud.

Vision simple d'une adresse :



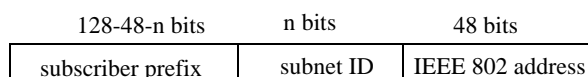
Vision plus évoluée d'une adresse :



#### 3.4.2 Exemples de vision d'adresses unicasts

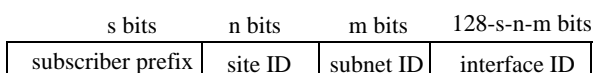
Utilisation des adresses MAC (IEEE 802)

- . similaire à IPX.
- . facilite l'autoconfiguration.



Adressage à hiérarchies multiples :

- . un organisme,
- . possède plusieurs sites,
- . sur lesquels il y a plusieurs réseaux (locaux),
  - l'organisation interne doit être souple.
  - (le préfixe de) l'adresse de l'organisme est unique.



### 3.4.3 Adresses utilisant le préfixe réservé : 0000 0000

Adresse non-spécifiée :

- indique l'absence d'adresse.
- peut être utilisé lors de l'initialisation d'un noeud.
- 0:0:0:0:0:0:0

Adresse "Loopback" :

- émission/réception par le même noeud.
- test.
- FE00:0:0:0:0:0:0:1

Adresses IPv6 compatibles avec IPv4.

. IPv4-compatible IPv6 address :

- 0:0:0:0:0:FFFF:<IPv4-address> avec l' @IPv4 en format decimal pointé
- pour les noeuds à la fois IPv4 et IPv6.
- permet d'utiliser facilement la technique du "tunnelling" (à travers un réseau IPv4) pour transmettre les paquets IPv6.

. IPv4-mapped IPv6 address :

- 0:0:0:0:0:0:<IPv4 address>
- pour les noeuds uniquement IPv4.

### 3.4.4 Adressage global attribué par les fournisseurs

. Registry ID :

- identifie les organismes d'enregistrements.
- ces organismes attribuent l'espace d'adressage aux fournisseurs.

. Provider ID :

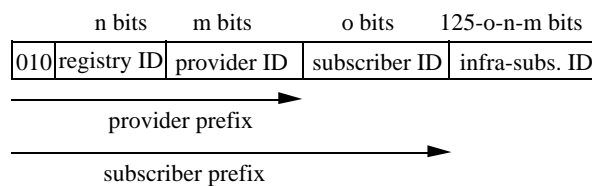
- identifie les fournisseurs au sein d'un organisme d'enregistrement.
- ces fournisseurs attribuent l'espace d'adressage aux abonnés.

. Subscriber ID :

- identifie les abonnés d'un fournisseur.
- ces abonnés gèrent leur sous-espace d'adressage

. Ce sous-espace peut lui même être organisé comme IPv4:

- subnet ID + host ID.



### 3.4.5 Adressage attribué localement

. Les routeurs ne doivent pas retransmettre des paquets munies de telles adresses.

. Adressage local au niveau d'une liaison (réseau local)

10 bits	n bits	118-n bits
1111111010	0...0	interface ID

. Adressage local au niveau d'un site :

- pour les organismes non encore interconnectés.
- prévoir un plan d'adressage compatible.

10 bits	n bits	m bits	118-n-m bits
1111111011	0...0	subnet ID	interface ID

### 3.5. Adresses anycasts

. Un paquet ayant une adresse anycast est routé vers **la station (l'interface) la plus proche** ayant cette adresse.

. Les adresses anycast sont syntaxiquement des adresses unicasts :  
- parmi l'espace d'adressage unicast.

. Le préfixe commun à un ensemble d'adresses unicast d'interfaces associées à la même adresse anycast définit topologiquement une région:

- cela permet de faciliter la gestion de routage (par région).
- si le préfixe est nul, la région n'existe pas, l'optimisation n'est pas possible.

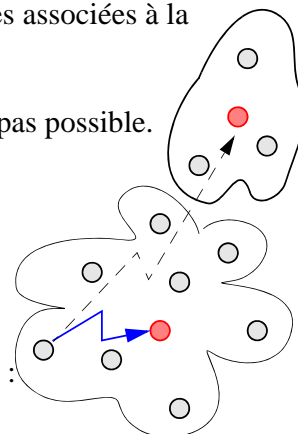
**! Attention :**

- ne doivent pas être utilisées comme des adresses d'émission.
- ne doivent pas être utilisées comme des adresses de stations.

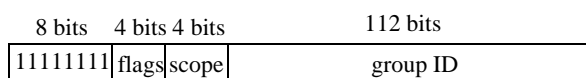
. Identification de services :

- pour atteindre un fournisseur (de services) spécifique,  
par exemple, pour atteindre le routeur du sous-réseau local :

n bits	128-n bits
anycast prefix	0...0



### 3.6. Adresses multicasts



. **Flags** (4 bits) : 000T

- T=0 : attribution permanente de l'adresse multicast
- T=1 : attribution non-permanente de l'adresse multicast

. **Scope** (4 bits) : limite la diffusion

- 0, F : réservé
  - 3, 4, 6, 7, 9, A, B, C, D: non utilisé
  - 1 : étendue restreinte au noeud
  - 2 : étendue restreinte à la liaison
  - 5 : étendue restreinte au site
  - 8 : étendue restreinte à l'organisme
  - E : étendue mondiale
- => le champ TTL était utilisé à cette fin par IPv4.

#### 3.6.1 Adresses multicast pré-attribuées

##### Exemples

- . Aucune station n'appartient à ce groupe réservé :
  - FF0s::0
- . Tous les noeuds (stations + routeurs) :
  - format général : FF0s::1 avec s={1, 2, 5, 8, E}
  - Exemple : toutes les interfaces de la liaison = FF02::1
- . Toutes les stations :
  - format général : FF0s::2 avec s={1, 2, 5, 8, E}
  - Exemple : toutes les interfaces (de type station) du noeud = FF01::2
- . Tous les routeurs :
  - format général : FF0s::3 avec s={1, 2, 5, 8, E}
  - Exemple : tous les routeurs du site = FF05::3
- . Tous les serveurs NTP :
  - format général : FF0s::43 avec s={1, 2, 5, 8, E}

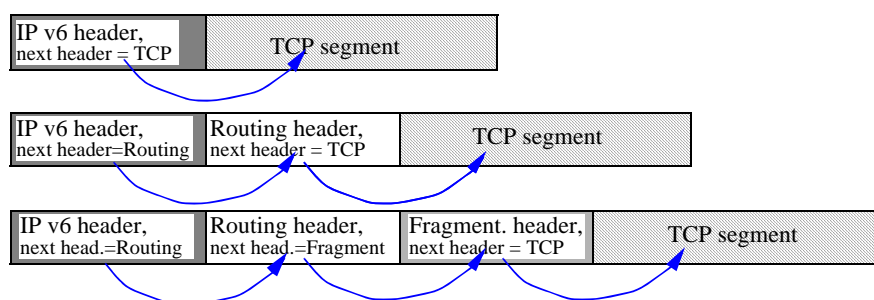
##### But

- . Optimisation des algorithmes de découverte :
  - minimisation de l'exploration.

## 4. Les options

### 4.1. Les entêtes optionnelles

- . La partie fixe de l'entête peut être suivie par un nombre quelconque d'entêtes optionnelles.
- . Chaque entête (optionnelle ou fixe) contient un champ "Next header".
- . Un **chaînage entre les entêtes** est établie à travers le champ "Next header".
- . La dernière entête décrit le type de protocole chargé du champ de données qui la suit. Dans ce cas le champ "Next Header" correspond exactement au champ "Protocol Type" de IPv4.



### 4.2. Le code des entêtes optionnelles

- . Hop-by-hop Options header : 0
- . Routing header : 43
- . Fragmentation header : 44
- . Authentification header : 51
- . Encapsulated Security Payload header : 52
- . No Next Header header : 59
- . Destination Options header : 60

and

- . ICMP : 2 (IPv6 !) (1=IPv4 !)
- . IGMP : 2 !
- . IP (in IP) : 4 !
- . TCP : 6
- . UDP : 17
- . ISO-TP4 : 29
- . ISO-CLNP : 80
- . IGRP : 88
- . OSPF : 89
- . etc.

### 4.3. L'ordonnement des entêtes optionnelles

=> Les options peuvent être traitées :

- (a) soit par tous les noeuds intermédiaires et le(s) destinataire(s) final,
- (a') soit par tous les destinataires intermédiaires et le(s) destinataire(s) final,
- (b) soit uniquement par le(s) destinataire(s) final.

=> L'exploitation d'une entête dépend du traitement de l'entête qui le précède :

- le résultats des traitements peuvent dépendre de leur ordre.

Ordre :

- . IPv6 header (a)
- . Hop-by-hop options header (a)
- . Destination options header (a')
- . Routing header (a')
- . Fragmentation header (b)
- . Authentification header (b)
- . Encapsulated security payload header (b)
- . Destination options header (b)
- . Upper layer header and data (b)  
ou No next header header (b)

### 4.4. Le traitement des entêtes

. La taille des entêtes optionnelles est exprimée en unités de 8 octets.

=> parallélisation des accès aux données

. Lorsqu'un noeud doit traiter l'entête suivante et que la valeur de son Next Header est non reconnue

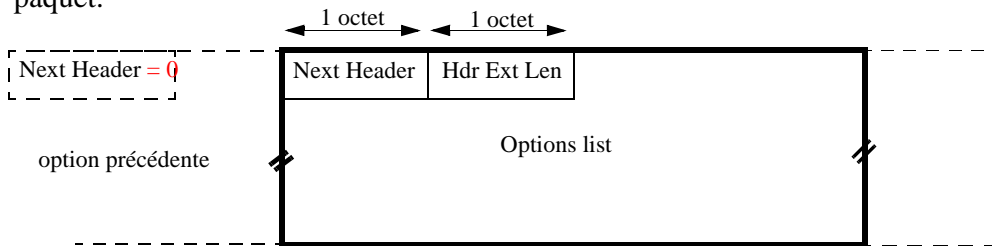
=> le noeud émet un paquet ICMP vers la source.

- ICMP code = 2 ("unrecognized Next header type encountered")
- ICMP pointer = la valeur non-reconnue.

### 4.5. L'entête "Hop-by-hop options"

**Next Header = 0.**

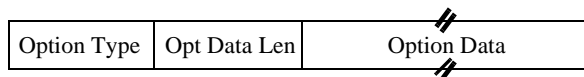
=> Informations examinées par chaque noeud situé sur le chemin du paquet.



- . Next Header (1 octet) : identifie la prochaine entête.
- . **Hdr Ext Len** (1 octet) : longueur totale de l'entête optionnelle Hop-by-hop (moins les 8 premiers octets) en multiples de 8 octets.  
=> nécessite un bourrage (padding option)
- . **Options list** (variable) : champ contenant une ou plusieurs options.

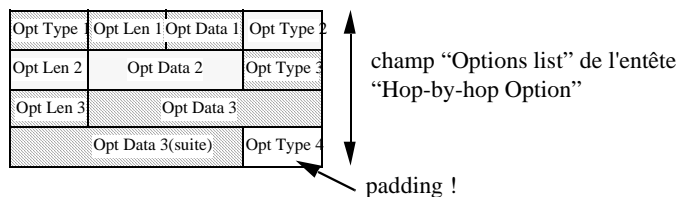
#### 4.5.1 Les options de l'entête "Hop-by-hop options"

. Chaque option est codée suivant le format **TLV** (type, longueur, valeur)



- **Option Type** (1 octet) : identifie le type de l'option
- **Opt Data Len** (1 octet) : longueur en octets du champ Option Data.
- **Option Data** (variable) : les informations spécifiques de l'option.

. Les options se succèdent toutes dans le champ Options de l'entête optionnelle Hop-by-hop Option.



### 4.5.2 Le sous-champ "Option Type"

Si l'option n'est pas reconnue (2 bits) :

- . Option type == 00xxxxxx : passer à l'option suivante.
  - . Option type == 01xxxxxx : détruire le paquet.
  - . Option type == 10xxxxxx : détruire le paquet, envoyer un message ICMP type 4 code 2.
  - . Option type == 11xxxxxx : détruire le paquet, envoyer un message ICMP type 4 code 2.
- ssi l'adresse de destination n'est pas une adresse multicast.

Modification du champ Option Data au cours de la transmission du paquet :

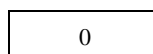
- . Option type == xx0xxxxx : pas de modification.
  - . Option type == xx1xxxxx : modification possible.
- => utilisé lorsque l'entête optionnelle d'authentification est présente : les entêtes susceptibles d'être modifiées sont considérées comme ayant des octets nuls.

### 4.5.3 Les options d'alignements

- . Les options d'alignements : Pad1 et Padn.
- . Communes aux entêtes ayant des options (entête Hop-by-hop, entête Destination)

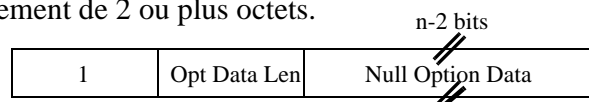
**Pad1** : option type =0

- alignement d'1 octet.
- cas spécial sans champ de longueur ni de valeur.



**Padn** : option type =1

- alignement de 2 ou plus octets.

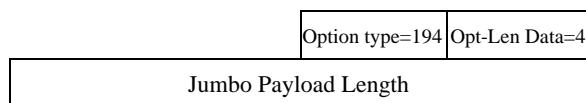


- Opt Data Len (1 octet) :
  - . bourrage de n octets => Opt Data len = n-2
- Null Option Data (variable) : n-2 octets nuls.



## 4.5.4 L'option "Jumbo payload"

## . Option type =194



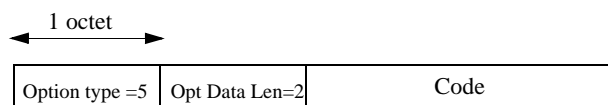
- Pour des paquets de données dont la taille dépasse 65535 octets (et <4 Go !).
- Requiert un alignement en frontière de  $4n+2$ .
- **Jumbo Payload Length** = longueur du paquet en octets à l'exclusion de l'entête IPv6 mais en incluant les entêtes optionnelles y compris l'entête Hop-by-hop.
- Le champ Payload Length de l'entête IPv6 est mise à zéro.
- La présence de l'option Jumbo Payload de l'entête Hop-by-hop exclue celle d'une entête Fragmentation.

- . Option invalide (longueur) ou incompatible (entête Fragment) :  
=> message ICMP code 0 (Parameter problem)

## 4.5.5 L'option "router Alert"

## . Option type =5

Rôle : Le paquet IP doit être traité spécifiquement par chaque routeur  
Défini par le rfc2711



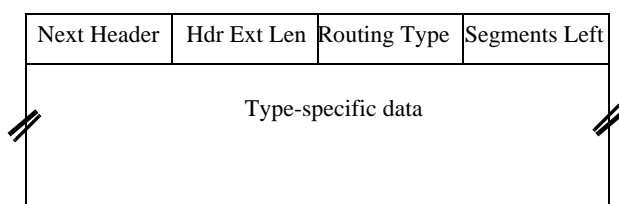
Code :

- 0 = message MLD
- 1 = message RSVP
- 2 = message "active network"
- etc.

## 4.6. L'entête Routing

Next header = 43.

=> informations examinées par chaque destinataires intermédiaires.



- . Next Header (1 octet) : identifie la prochaine entête.
- . Hdr Ext Len (1 octet) : longueur totale de l'entête optionnelles Routing (moins les 8 premiers octets) en multiples de 8 octets.
- . Routing Type (1 octet) : identifie un format spécifique pour les données.
- . Segment Left : nombre de destinataires intermédiaires restant à atteindre pour parvenir au destinataire final.
- . Type-specific data (variable) : champ contenant la liste des destinataires intermédiaires, c-à-d la route à suivre par le paquet.

### 4.6.1 L'entête Routing de type 0

- . Le champ Routing Type = 0 (le seul existant pour l'instant).
- . Similaire à l'option Source Routing de IPv4.

Next Header	Hdr Ext Len	Routing Type=0	Segments Left
Reserved	Strict/Loose bit map		
Address[1]			
Address[2]			
...			
Address[n]			

- . Hdr Ext Len (1 octet) : 2 fois le nombre de champs Address(<= 46)
- . Segments left (1 octet) : nombre de noeuds intermédiaires restant à visiter (<= 23 ! : ceci est controversé).
- . Strict/loose bit map (3 octets) :
  - à chaque bit correspond l'adresse de même numéro.
  - précise si l'adresse est celle d'une station qui doit être adjacente à la station précédente (1 : strict routing) ou non (0 : loose routing).

- . Pas d'adresse multicast.
- . Le bit de poids faible du champ Bit Map précise le routage (lâche ou strict) vers le noeud correspondant au champ Destination Address de l'entête IPv6.
- . Plusieurs entêtes Routing peuvent se suivre :
  - => spécification de chemins aussi longs que voulus.

Exemple :

Champs :	S -->	I1 -->	I2 -->	I3 -->	D
Source Address	S	S	S	S	S
Destination Address	I1	I2	I3	D	D
Header Length	6	6	6	6	6
Segments Left	3	2	1	0	0
Address [1]	I2	I1	I1	I1	I1
Address [2]	I3	I3	I2	I2	I2
Address [3]	D	D	D	I3	I3

#### 4.7. L'entête Fragment

**Next header = 44.**

=> Transmettre des paquets plus grands que le MTU du chemin.

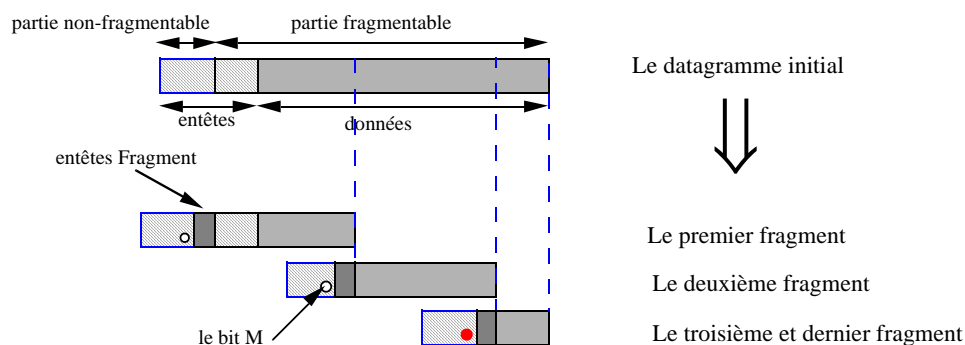
- La segmentation n'a lieu qu'à l'émetteur et le réassemblage n'a lieu qu'au récepteur.

Next Header	Reserved	Fragment Offset	Res	M
Identification				

- . Next Header (1 octet) : identifie la prochaine entête.
- . Reserved (1 octet) : initialisé à 0, ignoré au récepteur.
- . **Fragment Offset** (13 bits) : le déplacement (en unités de 8 octets) des données suivant cette entête relativement au début de la partie fragmentable du paquet initial.
- . Res (2 bits) : 00.
- . **M** (1 bit) :
  - 0 = dernier fragment,
  - 1 = il existe des fragments du même paquet qui suivent.
- . **Identification** (4 octets) : identifie tous les fragments appartenant au même paquet initial (unique si couplé avec l'@ de la source). Similaire au même champ (de 16 bits) d'IPv4.

## 4.7.1 Constitution des fragments

- . Au sein du paquet initial, on distingue 2 parties :
  - celle non-fragmentable, contient toutes les entêtes précédant l'entête Routing (y compris celle-ci, si elle existe).
  - celle fragmentable, le reste du paquet initial (les autres entêtes et les données).



- . Procédé assez proche d'IPv4.

- Tous les fragments d'un même paquet initial ont les mêmes entêtes issues de la partie non-fragmentable de ce paquet,
- Sauf pour l'entête IPv6 :
  - . elle conserve tous ses champs,
  - . sauf le champ Payload length qui contient la longueur utile de son fragment.
- Une entête Fragment est ajoutée à tous les fragments :
  - . en assurant le chaînage des champs Next Header avec les entêtes précédentes et suivantes.
  - . en mettant à jour les champs Fragments Offset, M.
  - . leur champ Identification contiennent tous la même valeur.
  - . placé entre la partie non-fragmentable et la portion fragmentée.

#### 4.7.2 Réassemblage des fragments

=> Le réassemblage n'a lieu qu'au destinataire final.

- . Le premier fragment :
  - contient un champ Fragment Offset =0.
  - permet la restitution de la suite d'entêtes du paquet initial
- . Le dernier fragment :
  - contient un bit M = 0.
  - permet de recalculer la taille utile du paquet initial.
- . Calcul de la longueur utile du paquet initial.
- . Calcul et vérification que tous les fragments sont présents.
- . Reconstitution du paquet initial.

=> la longueur utile du paquet initial < 65535 octets.

#### 4.7.3 Taille des paquets

- . Le **MTU** des liaisons IPv6  $\geq 576$  octets :
  - la couche inférieure doit posséder une fonction de fragmentation /réassemblage spécifique pour assurer ce service, si nécessaire.
- . Si un noeud veut utiliser une longueur supérieure à 1500 octets, il doit auparavant s'être assuré que le destinataire le tolère.

Les noeuds IPv6 doivent utiliser la technique de recherche du MTU de chemin :

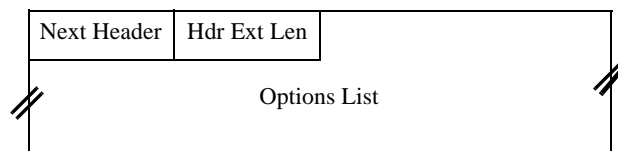
- rfc 1191 : technique par essai/erreur
- utilisée par IPv4.
- le "Do Not Fragment" bit est implicite dans IPv6.
- pour IPv6 le message ICMP "Datagram too big" indique la valeur exacte du MTU.

## 4.8. L'entête Destination Options

**Next header = 60**

=> contient les informations qui doivent être analysées par le (ou les) destinataires final (aux).

- . l'inverse de l'entête Hop-by-hop Options, mais le format est similaire.
- . permet de minimiser le nombre d'entêtes optionnelles.



- . Next Header (1 octet) : identifie la prochaine entête.
- . Hdr Ext Len (1 octet) : longueur totale de l'entête optionnelle.
- . Options List (variable) : contient une ou plusieurs options.
  - les seules options actuellement normalisées sont Pad1 et Padn !
  - celles utilisées par l'entête Hop-by-hop Options.

## 5. La sécurité sur IPv6

IPv6 introduit deux services de sécurité :

- L'authentification+ (rfc 4302)
  - La confidentialité (rfc 4303)
  - . L'authentification+ permet de certifier :
    - l'émetteur du message, et autres infos générales (son authentification),
    - le contenu du message (son intégrité).
  - . La confidentialité assure que les infos protégées ne seront pas lues par des tiers.
  - . Ces deux services sont basés sur des techniques de chiffrement
    - un algorithme de chiffrement (public) + un secret (clef de sécurité)
      - algorithme symétrique : une clef partagée
      - algorithme asymétrique : une clef privée + une clef publique.
  - . L'émetteur et le(s) récepteur(s) partagent une association de sécurité :
    - contexte de sécurité (l'algorithme, les paramètres, la clef, etc.)
    - un identificateur de ce contexte (Security Association identifier)
- Ces services ne sont mis en oeuvre que par les noeuds d'extrémité (surcoût). Notamment par IKE (protocole 500, rfc 4306).

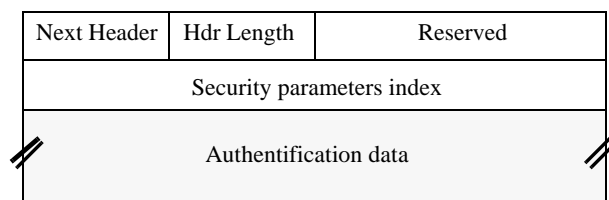
L'authentification+ et la confidentialité sont deux services qui sont repris par IPv4 (protocole 51 et 50, resp.).

### 5.1. L'entête Authentification

. **Next Header = 61**

. Principe :

- une signature est calculée à partir des infos grâce à un procédé "secret".
  - l'émetteur place cette signature dans le paquet avec les données.
  - le récepteur vérifie que la signature reçue correspond à la signature re-calculée à partir des données du paquet reçu.
- . Le procédé de calcul et les données sur lesquelles il s'applique sont libres (par ex. HMAC-MD5 - "Hash Message Authentication Code-message digest algorithm").
- . Attention aux champs qui sont modifiées par les routeurs !



Next header : le code du prochain header (normalement TCP).

**Hdr length** : la longueur du champ Authentication data en mots de 32 bits

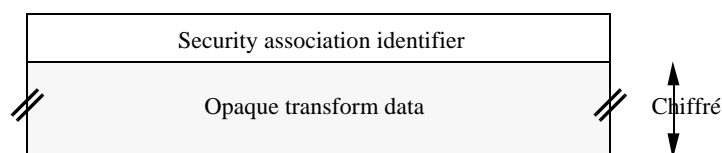
l'entête s'adapte à toutes les longueurs de signature.

**Security parameters index** : identifie le contexte.

**Authentication data** : la signature (clef d'authentification).

## 5.2. L'entête de chiffrement

- . Next Header = 62
- . L'entête "Encapsulating Security Payload"
- . La totalité de cette entête est encodée sauf les premiers 32 bits
- . Le procédé d'encodage peut être :
  - un procédé de chiffrement (RC2, PGP, DES, RSA, ITU X.509, etc),
  - un procédé de compression (LZ77, LZS, etc).
- . C'est **toujours le dernier entête** (pas de champ Next Header) !

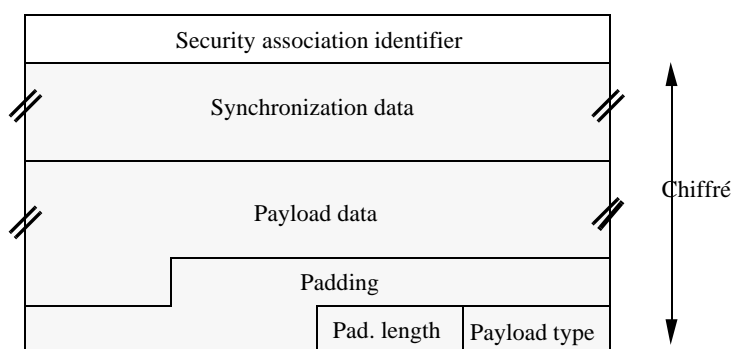


- . Security association identifier :
  - identifie le contexte de sécurité (idem SPI)
- . Opaque transform data :
  - les données encodées
  - le format dépend de l'encodeur choisi

### 5.2.1 Exemple l'entête ESP et le chiffrement DES

Exemple : DES-CBC (Data Encryption Standard- Cypher Block Chaining) par défaut.

- . Synchronization data :
  - données initiales pour le procédé de calcul
  - par un générateur aléatoire
  - rendre + difficile le décodage
- . Payload data : les données chiffrées
- . Padding : le bourrage
- . Payload type : précise le type des données (par ex. TCP)





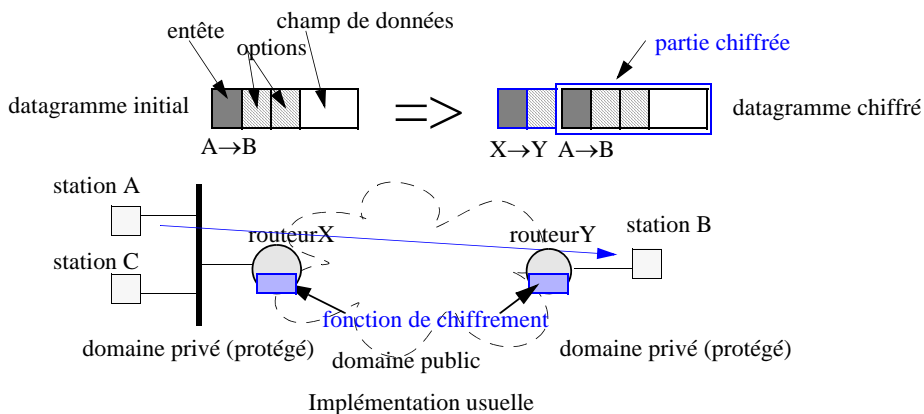
### 5.2.2 Modes d'utilisation

Deux modes d'utilisation de l'entête ESP sont prévus :

- . le **Tunnel-mode** et le **Transport-mode**

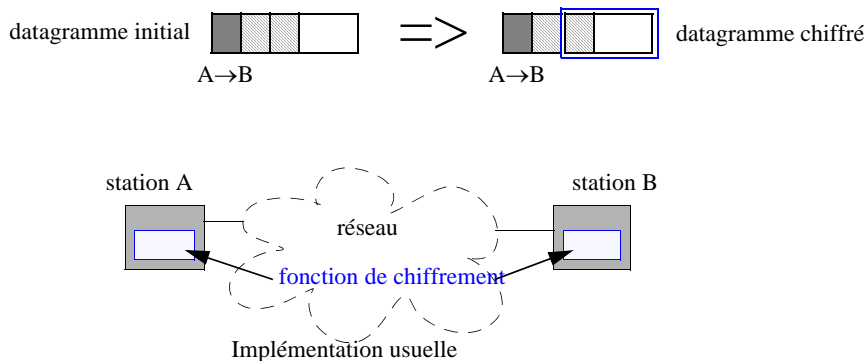
. **Tunnel-mode** :

- l'entête ESP encapsule la totalité d'un datagramme IP
- assure une confidentialité totale :  
=> même des informations situées dans l'entête du datagramme initial.
- des serveurs de chiffrement sont souvent utilisés



. **Transport-mode** :

- l'entête ESP encapsule une partie (entêtes terminales de données) d'un datagramme IP
- moins coûteux.
- la fonction de chiffrement est souvent interne aux stations d'extrémité



## 6. Conclusion

### Optimisation :

- . traitement : chaînage d'entête optionnelles, suppression du "checksum" et du reste
- . acheminement ("forwarding") : les entêtes Hop-by-hop options ou Destinations

### Augmentation :

- . l'adressage : adressage sur 16 octets, adressage anycast, plans d'adressage variés
- . la sécurité : authentification, confidentialité
- . la gestion de groupe : intégration
- . la mobilité, l'autoconfiguration (IPv4 DHCP-"plug and play")
- . la qualité lors du transfert de données : flow label + RSVP

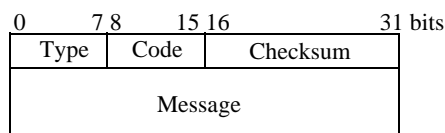
### Conséquences :

- . sur les autres protocoles : ICMP+, IDRP, OSPF, RIP+, DNS+
- . sur l'interface de programmation (par ex. : WinSock II)
- . la transition : "tunnelling" + adressage compatible => **6bone**
- . "dual stack system"

## 6.1. Evolution de ICMP

### 6.1.1 Présentation

- . "Internet control message protocol"
- . Le protocole indispensable au fonctionnement d'IP
- . Similaire à ICMP pour IPv4.
- . Augmentation des fonctionnalités :
  - gestion de groupe : IGMP ("Internet Group Membership Protocol")
    - . gestion fine du groupe : dépendante de la source
  - "server discovery"
  - "neighbor discovery"



## 6.1.2 Nouveaux types et codes d'ICMP

1 Destination unreachable	0 No route to destination	
	1 Route administratively prohibited	
	2 Address unreachable	
	3 Port unreachable	
2 Packet too big		return : MTU
3 Time exceeded	0 Hop limit exceeded	
	1 Reassembly time exceeded	
4 Parameter problem	0 Erroneous header	
	1 Unrecognized Next header type	
	2 Unrecognized Option	
128 Echo request		
129 Echo reply		
130 Group membership query		
131 Group membership report		
132 Group memb. terminaison		
133 Router sollicitation		
134 Router advertisement		
135 Neighbor sollicitation		
136 Neighbor advertisement		
137 Redirect		