

# Le protocole IP

(Z:\Enseignements2007-2008\REPR\Cours\08.IP.fm- 30 janvier 2007 15:02)

## PLAN

- Introduction à IP
- L'acheminement
- L'adressage
- La segmentation
- Le datagramme
- L'acheminement (suite)
- Conclusion

## 1. Introduction au protocole Internet

### 1.1. Présentation

- IP** : “Internet protocol”
- . Rfc 791 (standard, required)
  - . version 4 (septembre 1981)

#### Transmission de données :

- par paquet ⇒ “datagram”
- en mode non connecté (sans contexte : simple mais sans mécanisme de contrôle!)
- transparente
- “best effort”

#### Interconnexion :

- adressage
- acheminement

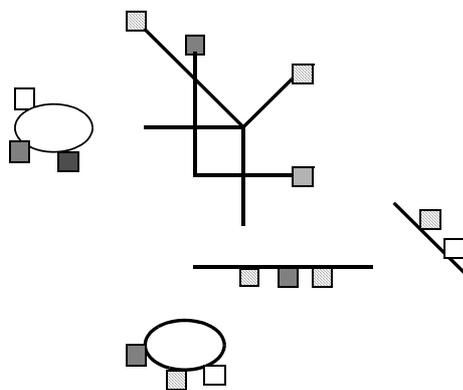
#### Homogénéisation :

- hétérogénéité des stations, de leur localisation, des méthodes d'accès, des topologies, etc.)
- segmentation (fragmentation)
- minimum commun : simple, allégé, performant

## 1.2. Les équipements

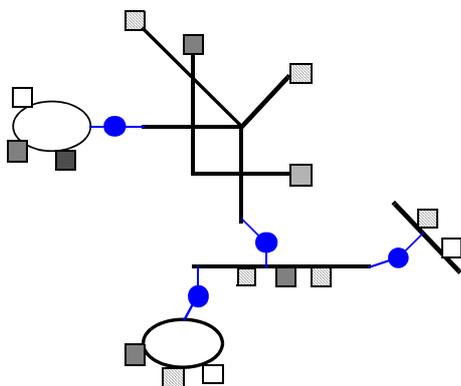
Des stations hétérogènes, 

sur des infrastructures hétérogènes : Ethernet, FDDI, réseau ATM, etc...



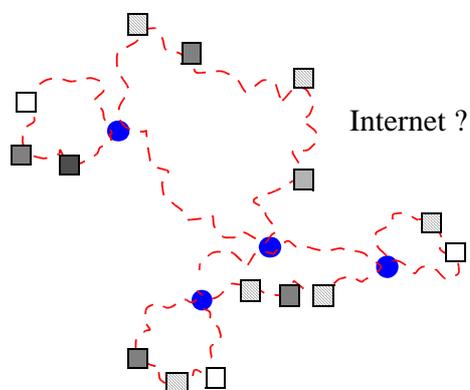
## 1.3. Les routeurs

Interconnexion grâce à des routeurs (équipements d'interconnexion) : 



#### 1.4. L'hétérogénéité des sous-réseaux

IP masque l'hétérogénéité des (sous-)réseaux :

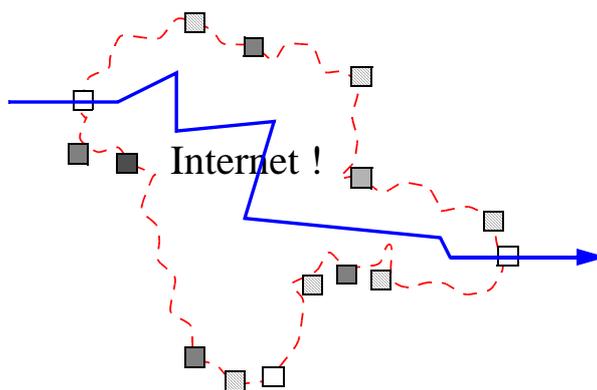


**Internet = réseau de réseaux !**

on remarque que les routeurs appartiennent à plusieurs (sous-)réseaux

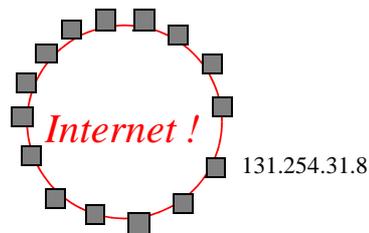
#### 1.5. L'hétérogénéité de la localisation

IP assure la transmission des données, masque les routeurs et les sous-réseaux :



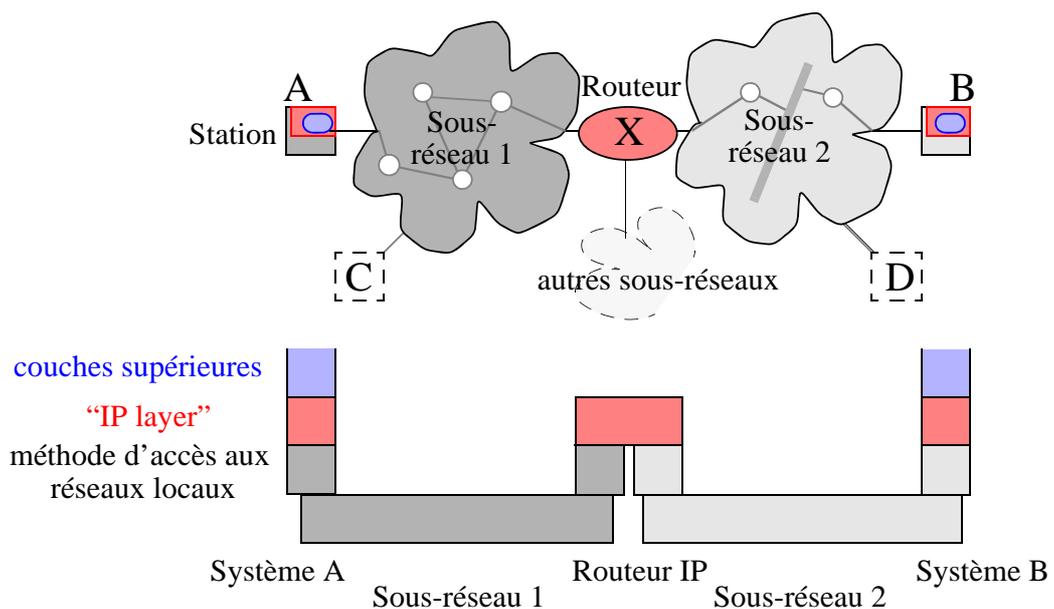
### 1.6. L'Internet idéal

IP masque la localisation et le type des stations, notamment grâce à l'adressage IP : 131.254.31.8.

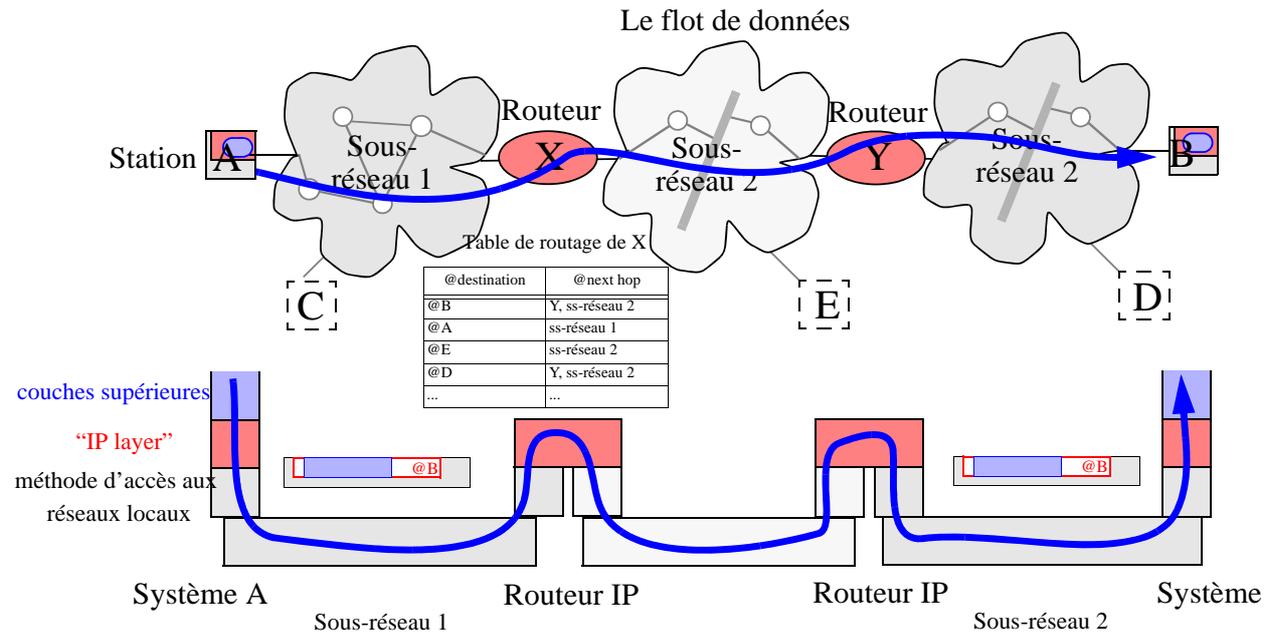


## 2. L'acheminement des datagrammes

### 2.1. Les acteurs

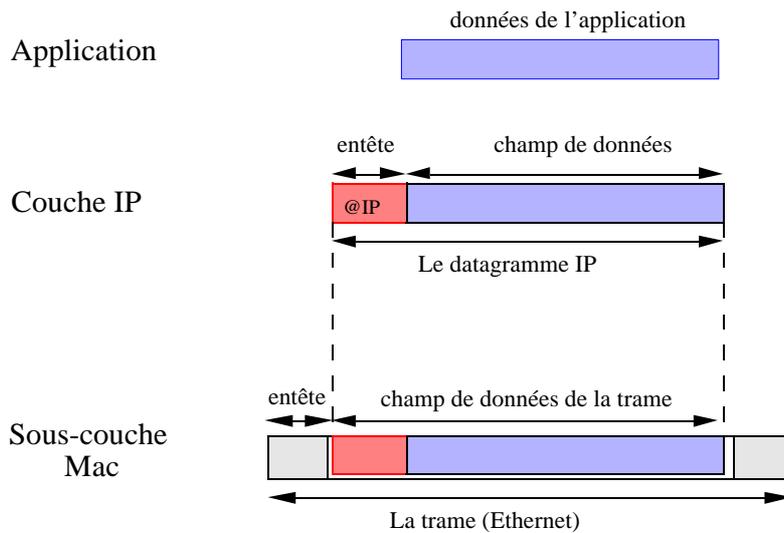


### 2.2. La table de routage



### 2.3. Le transport des datagrammes IP

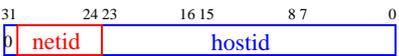
Les datagrammes sont transportés (encapsulés) dans les trames des réseaux sous-jacents



### 3. L'adressage d'Internet

#### 3.1. Les classes d'adressage

- . Hiérarchique :
  - identificateur du (sous-)réseau (**netid**)
  - identificateur de la station dans le (sous-)réseau (**hostid**)
- . Taille fixe (32bits)
- . 4 classes d'adressage (5!)

Classe :		Nombre maximum de sous-réseaux	Nombre maximum de stations/sous-réseaux	
unicast	A		$2^7$	$2^{24}$
	B		$2^{14}$	$2^{16}$
	C		$2^{21}$	$2^8$
	D			
	E			

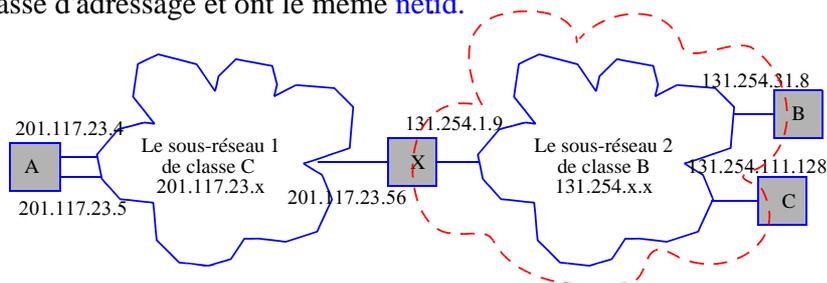
#### 3.2. La notation des adresses sous Internet

Notation décimale pointée (dotted decimal) :  $D_1.D_2.D_3.D_4$   
 par exemple : h2s.irisa.fr = 131.254.31.8

- $D_1 < 128 \Rightarrow$  classe d'adresse A
- $128 \leq D_1 < 192 \Rightarrow$  classe d'adresse B
- $192 \leq D_1 < 224 \Rightarrow$  classe d'adresse C

Attribution des adresses :

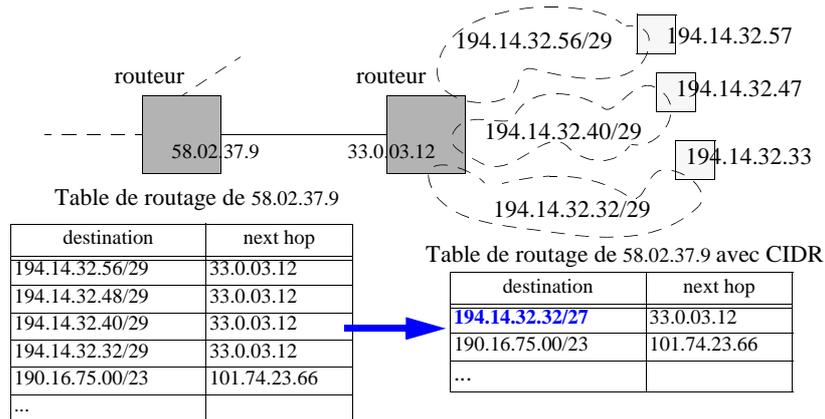
- . Une station IP doit avoir au moins une adresse IP unicast.
- . Une station peut avoir autant d'adresses que de connexions à un ou plusieurs (sous-)réseaux (appelé interface)
  - $\Rightarrow$  l'adresse identifie la connexion et pas la station !!!
- . Toutes les stations connectées à un même (sous-)réseau appartiennent à la même classe d'adressage et ont le même **netid**.



### 3.3. Le CIDR

- . “Classless Inter-Domain Routing” (rfc 1467, 1518, 1519)
- . Prévention de l'augmentation de la taille des tables de routage :
- . Chaque routeur route un paquet vers le prochain routeur en fonction du “longest prefix match” entre l'adresse de destination du paquet et les différentes adresses connues par le routeur.
- . Notation :
  - 131.254.26.13/22  $\Rightarrow$  le préfixe = 22 premiers bits de 131.254.24.00

Exemple d'agrégation :



## 4. Segmentation

### 4.1. Présentation

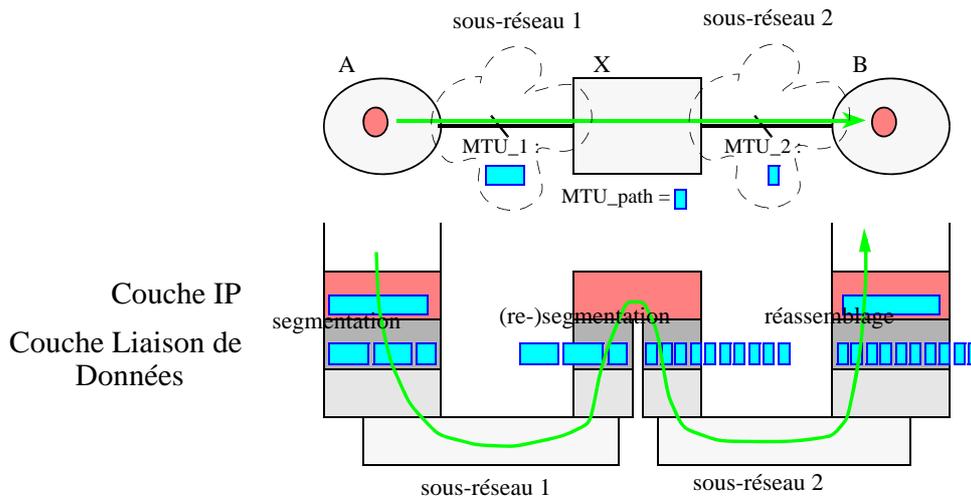
#### Problème :

- Adaptation de la taille des unités de données à transmettre à la taille des unités de transport.
- . L'utilisateur veut être capable de transmettre des unités de données de taille quelconque.
  - . Les (sous-)réseaux offrent des unités de transport de données de taille limitée :
    - Ethernet <1500 octets
    - FDDI < 4500 octets
    - X25.3 <= 128 octets (par défaut)
    - ATM = 48 octets (sans la sous-couche AAL)

#### Solution :

- . A l'émetteur :
  - découper l'unité de données à transmettre en segments de taille adaptée aux unités de transport (segmentation).
- . Au récepteur :
  - reconstituer l'unité de données initiale (réassemblage).
- . Aux routeurs :
  - si les sous-réseaux successifs comportent des unités de données de transport de tailles inférieures : (re-)segmentation.

## 4.2. Exemple

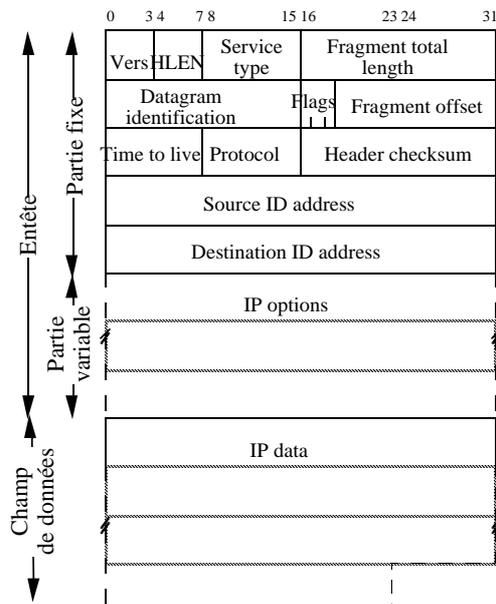


**MTU** : Maximum Transfer Unit

- . longueur maximum d'un datagramme IP pour qu'il puisse être transmis sans segmentation sur la liaison sous-jacente
- . MTU path : le MTU minimum pour l'ensemble des liaisons d'un chemin
- . MTU par défaut sous IPv4 = 576 octets

## 5. Le datagramme IP

### 5.1. Le format du datagramme IP



Les paquets de données d'IP ont pour nom **datagram**(me) :

⇒ télégramme : sans connexion

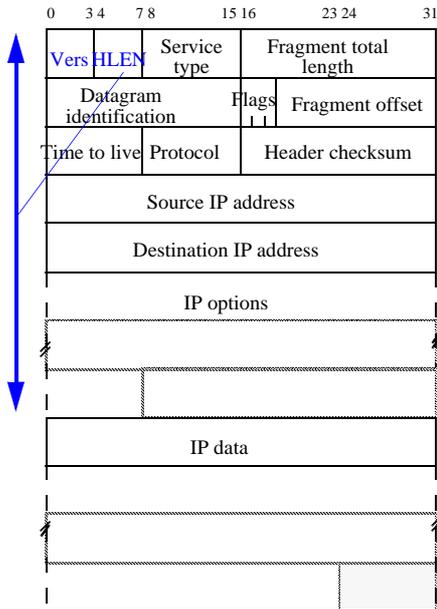
Structure d'un datagramme :

- Une entête :
  - . une partie de taille fixe,
  - . une partie de taille variable (les options),
- Un champ de données :
  - . de longueur variable.

Alignement sur des mots de 32 bits pour optimiser la vitesse de traitement.

La taille maximum d'un datagramme est de 64 Koctets ( $2^{16}$ ).

## 5.2. Les champs Vers et HLEN du datagramme IP



**Vers** (4 bits) : version du protocole.

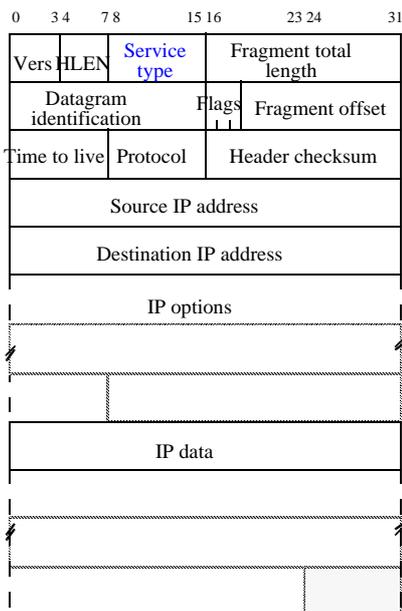
- . Version courante : 4 (septembre 1981)
- . Version nouvelle IPv6 : 6! (décembre 95)
- . Un même réseau peut accueillir différentes versions de protocole : les récepteurs écartent les datagrammes qu'ils ne peuvent pas décoder.

**HLEN** (4 bits) : "Header length"

- . Longueur de l'entête, (en mots de 4 octets) les options IP (de longueurs variables) comprises.
- . Au minimum, HLEN = 5 lorsqu'il n'y a pas d'options (longueur de la partie fixe de l'entête).
- . Longueur max. de l'entête = 60 octets

## 5.3. Le champ Type of Service du datagramme IP

### 5.3.1 Actuellement

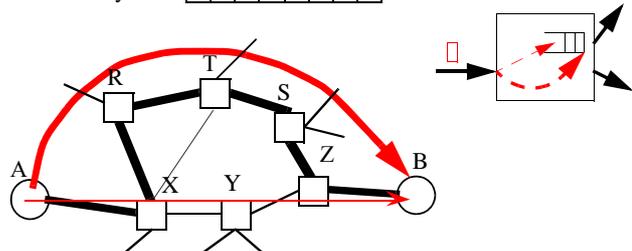


**Service type** (8 bits) : ToS "Type of service"

- . Qualifie le service de transmission demandé.
- . Utilisé pour optimiser l'algorithme de routage :
  - Priorités entre les différents types de flux de données.
  - Critères de choix lors du routage entre des chemins alternatifs.

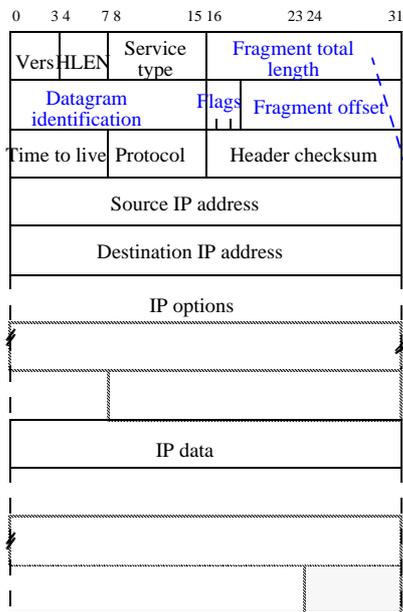
PRE : "precedence" = 0 ⇒ 7  
normale haute priorité

D : "delay"  
T : "throughput"  
R : "reliability"



## 5.4. La fragmentation

### 5.4.1 Les champs du datagramme IP pour la fragmentation

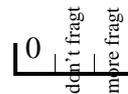


**Fragment total length** (16 bits) : longueur totale du fragment (et pas du datagramme !) en octets (< 64 koctet).

**Datagram identification** (16 bits) : Identification unique du datagramme et de tous ses fragments.

**Flags** (3 bits) :

- “do not fragment bit” : segmentation interdite  
=> MTU discovery (maximum transfer unit)
- “more fragment bit” : dernier fragment d'un datagramme.

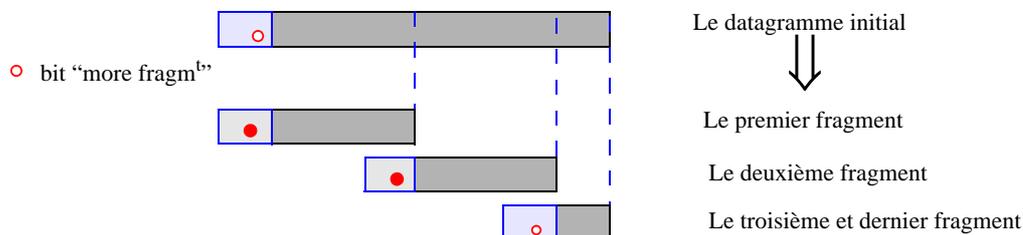


**Fragment offset** (13 bits) : déplacement relatif des fragments par rapport au début du datagramme (en unité de 8 octets).

### 5.4.2 Le traitement du datagramme

Constitution des fragments :

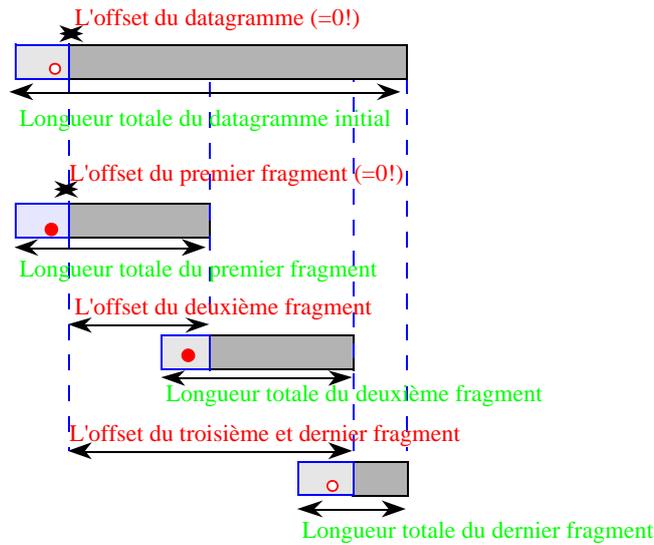
- Les entêtes des fragments comportent les mêmes champs et ont les mêmes contenus que ceux du datagramme dont ils proviennent :
  - même identification de datagramme, même type de service, même TTL, même protocole, mêmes adresses.
  - sauf pour les champs de longueur, le bit “More fragment” et le “Header checksum” !



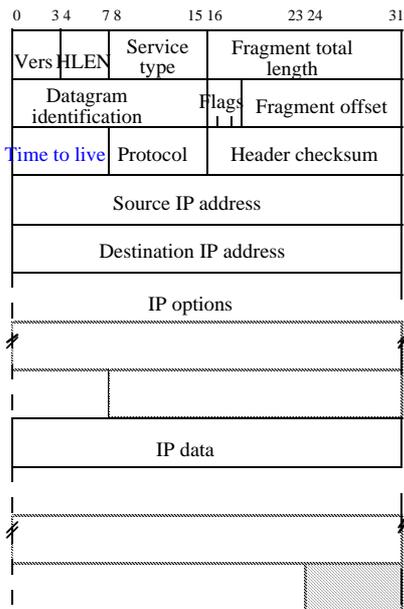
Réassemblage des fragments :

- Calcul de la taille totale du champ de données du datagramme :  
 $\text{offset}(\text{du\_dernier\_fragment}) * 8 + \text{longueur\_totale}(\text{du\_dernier\_fragment})$ 
  - longueur\_de\_l\_entête(du\_dernier\_fragment) \* 4.
- Calcul et vérification que tous les fragments sont présents.
- Reconstitution du champ de données du datagramme.

### 5.4.3 Offset et longueur



### 5.5. Le champ TTL



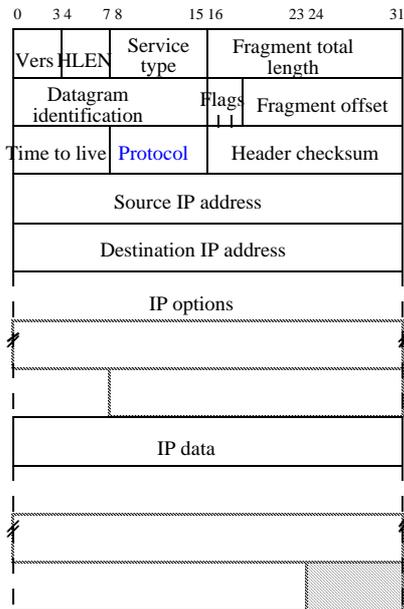
**Time to live** (8 bits) : durée de vie résiduelle du fragment (en nombre de "hops" ou de secondes de résidence).

- . valeur initialisée par l'émetteur.
- . décrétementée par chaque routeur et le récepteur,
- . si TTL=0 alors le datagramme est détruit, => on retourne à l'émetteur un message ICMP

Fonction :

- . limitation de l'étendue de diffusion des paquets ("scope").
- . suppression des trames fantômes.
- . limite la durée d'attente des fragments.

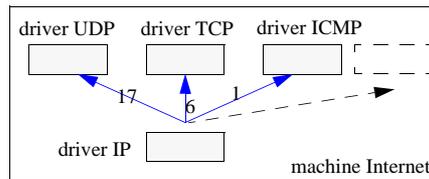
### 5.6. Le champ Protocol du datagramme IP



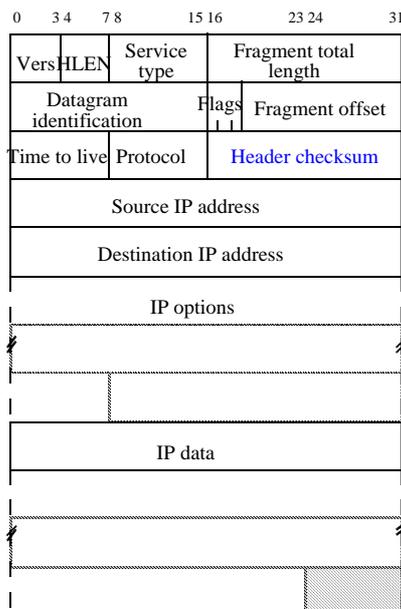
**Protocol (8 bits) :** identification du protocole chargé d'exploiter (décoder) le champ de données.

- . UDP = 17
- . TCP = 6
- . ICMP = 1
- . IP = 4 (IP in IP, par ex : mbone)
- . => RFC 1700

Enchaînement des drivers à la réception d'un message :



### 5.7. Le champ Checksum du datagramme IP



**Header checksum (16 bits) :**

Complément à 1 de la somme en complément à 1 des demi-mots (16 bits) constituant l'entête (options incluses) du fragment.

- . peu de calcul et calcul facile
- . capacité de détection faible

Permet la détection des erreurs

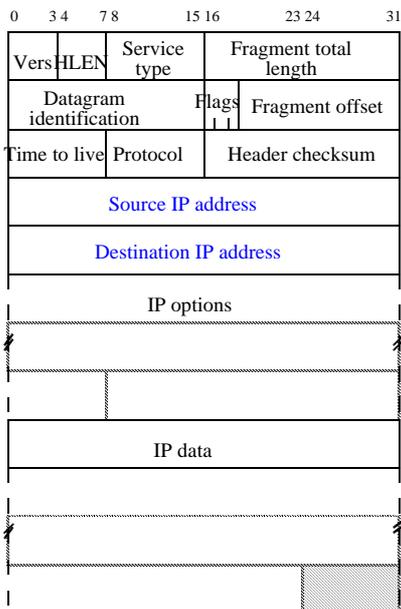
Pas de mécanismes de récupération d'erreur sous IP !

Procédure :

- le checksum est calculé à l'émetteur, puis placé dans le datagramme
- le checksum est vérifié à chaque réception du datagramme (routeurs intermédiaires et le destinataire final)
- s'il est incorrect le datagramme est simplement détruit

Exemple :  $\text{inv}(4500+05dc+e733+222b+ff11+0000+c02c+4d60+c02c+4d01)=\text{inv}(4\ 6e04)=\text{inv}(6e08)=91f7$

### 5.8. Le champ Address du datagramme IP

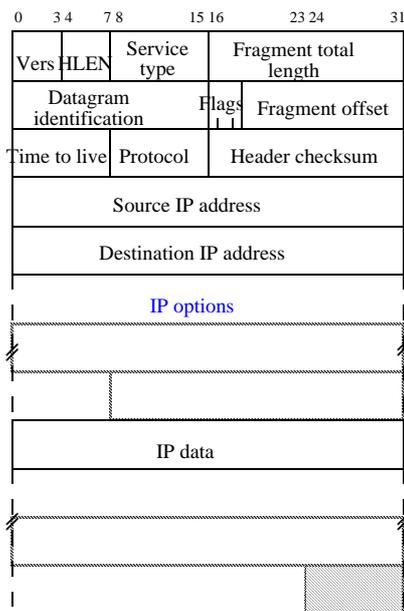


**Source IP address (32 bits) :**  
 identifie la station émettrice.  
 . retour à l'envoyeur (message ICMP).

**Destination IP address (32 bits) :**  
 identifie le récepteur (ou le groupe).  
 . permet l'acheminement jusqu'au(x) destinataire(s).

### 5.9. La partie variable de l'entête du datagramme

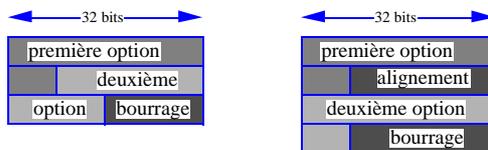
#### 5.9.1 Le champ *IP options* du datagramme IP



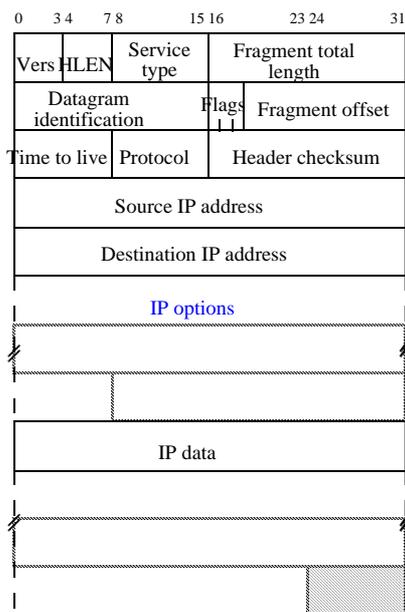
**Partie variable de l'entête**  
 . limitée à 20 octets  
**Leur traitement ralentit le routage :**  
 . certains routeurs ont deux files de datagrammes :  
 - datagrammes avec options  
 - datagrammes sans option  
**Fonction principale :**  
 . choix de la route par l'émetteur

**Assemblage des options :** indépendamment des mots de 32 bits ou non.

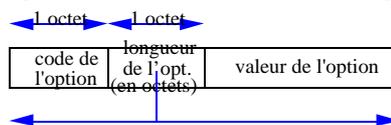
**Exemples de champ *IP options* :**



### 5.9.2 Les différentes options



Structure générale d'une option :  
type, longueur, valeur ("TLV encoding")



Code de l'option : 

7	6	5	4	0 bit
copy	class	number		

Quelques options :

classe	numéro	longueur	sémantique
0	0	-	"End of Option list" : bourrage de fin de liste d'options
0	1	-	"NOP" : alignement de fin d'option
0	2	11	"Security" : rfc 1108
0	3	>3	"Loose source routing"
0	7	>3	"Record route" -> trace_route
0	9	>3	"Strict source routing"
2	4	>4	"internet timestamp"

### 5.9.3 Les options Source Routing

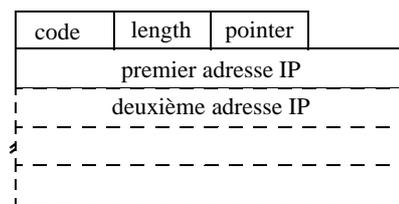
Un chemin = Liste des adresses IP des routeurs  
. 9 au maximum !

Record Route :

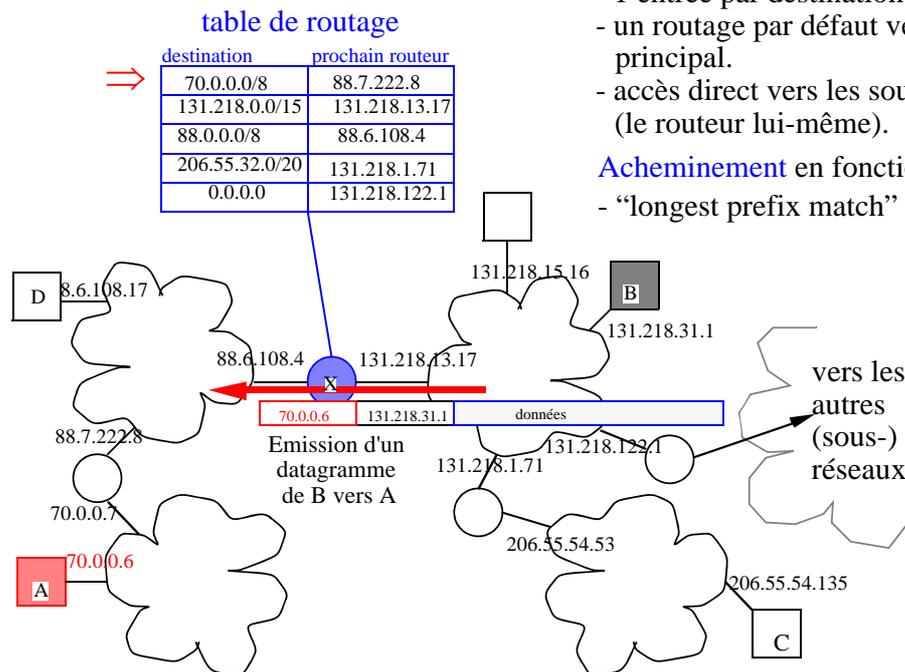
- . l'émetteur prépare une liste vide (pointer=3),
- . qui est remplie par chacun des routeurs du chemin (pointer+=4),
- . tant qu'il y a de la place dans la liste (pointer<length).

Source Routing :

- . le chemin que doit suivre le datagramme est explicitement décrit.
  - **strict source routing** : totalement décrit (S'il manque un routeur le datagramme est détruit).
  - **loose source routing** : partiellement décrit (S'il manque un routeur l'algorithme d'acheminement standard est appliqué).
- . à chaque routeur intermédiaire, le prochain routeur de la liste est échangé avec la destination, et le pointeur incrémenté.



## 6. L'acheminement du datagramme : "datagram forwarding"



- 1 entrée par destination connue
- un routage par défaut vers un routeur principal.
- accès direct vers les sous-réseaux locaux (le routeur lui-même).

Acheminement en fonction du :

- "longest prefix match"

## 7. Conclusion

Le protocole IP assure la transmission de données entre stations hétérogènes en utilisant un réseau formé par l'interconnexion de (sous-)réseaux hétérogènes :

- . acheminement,
- . adressage (adresse IP),
- . fragmentation.

Protocole simple (sans connexion):

- . détection des erreurs (de l'entête uniquement),
- . sans mécanisme de récupération des erreurs (perte, duplication, congestion).

Nécessite d'autres protocoles :

- + ICMP (erreur)
- + RIP (routage)
- + ARP (résolution d'adresse)
- + ...

Et **TCP** ou **UDP** !

### 7.1. Exemple de datagramme

Un datagramme IP constitué des octets suivants représentés en hexadécimal en lisant de gauche à droite et de haut en bas (par exemple : le 1er octet vaut  $49_{16}$ , le second :  $00_{16}$  et le cinquième :  $01_{16}$ ).

49	00	00	30
01	23	00	00
0D	11	42	DB
83	FE	3D	0D
CO	2C	4D	51
07	0F	0B	83
FE	0B	01	CO
2C	4D	4D	00
00	00	00	00
84	01	00	07
00	0C	00	00
01	02	03	04