

# Le protocole IP

(Z:\Polys\Internet RES0\5.IP.fm- 27 janvier 2016 15:55)

## 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 (et numérique)
- de type “best effort”


#### Interconnexion :

- adressage
- acheminement

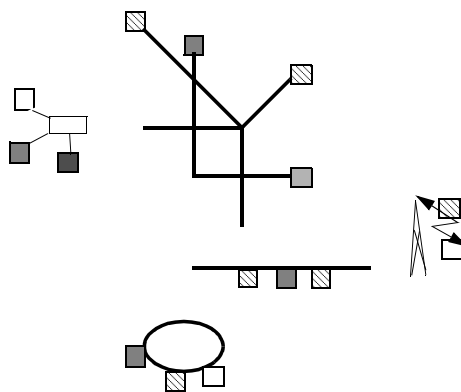
#### Homogénéisation :

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


## 1.2. Les équipements

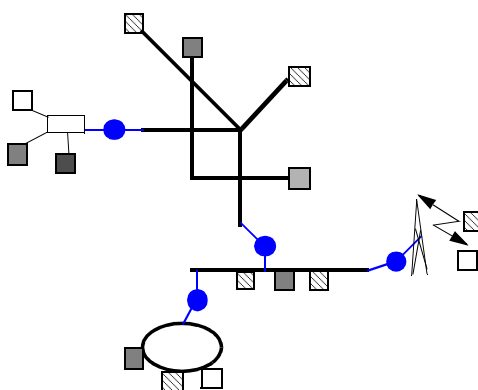
Des stations hétérogènes, 

sur des infrastructures hétérogènes : Ethernet, FDDI, WiFi, etc.



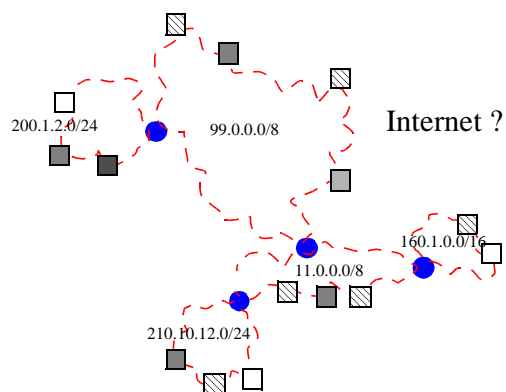
## 1.3. Les routeurs

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



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

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

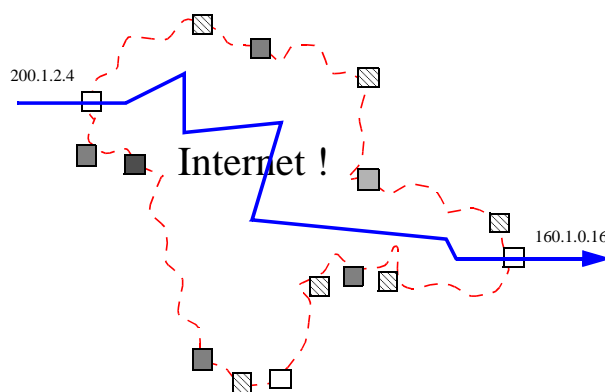


**Internet = réseau de réseaux !**

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

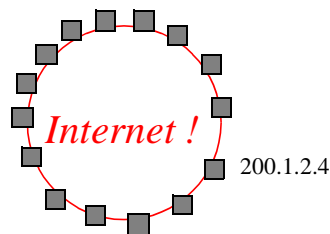
### 1.5. L'hétérogénéité de la localisation

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



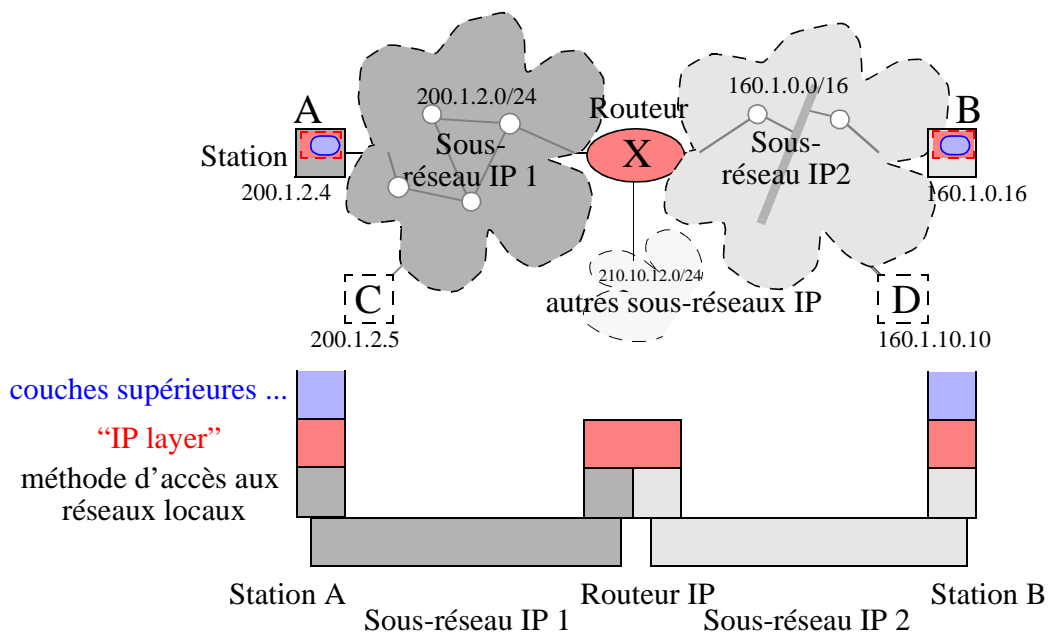
### 1.6. L'Internet idéal

IP masque la localisation et le type des stations, notamment grâce à l'adressage IP : 200.1.2.4.



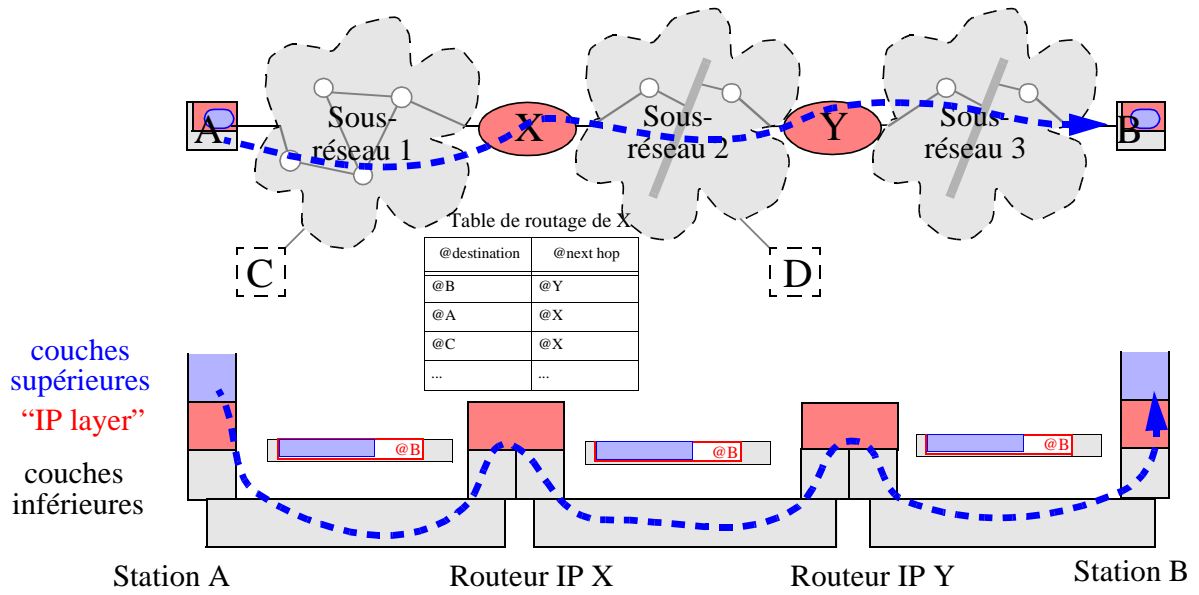
## 2. L'acheminement des datagrammes

### 2.1. Les acteurs



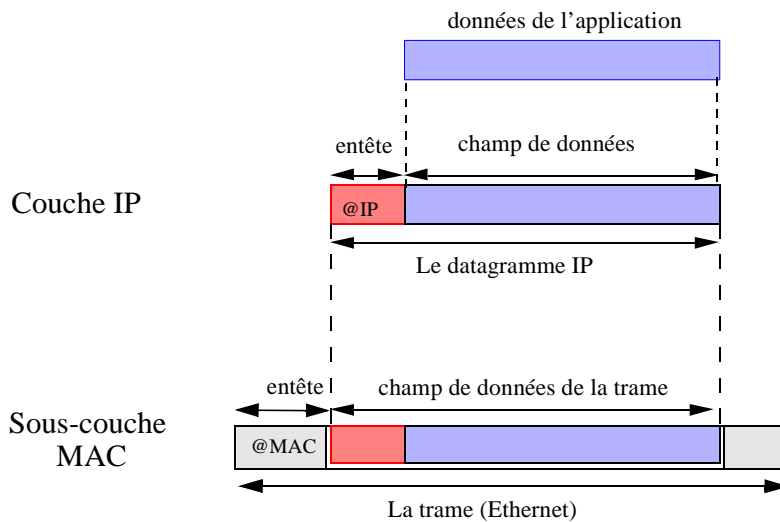
## 2.2. La table de routage

### L'acheminement du flot de données



## 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 adresses IP

Une adresse IP **identifie** l'interface de connexion à IP d'une station.

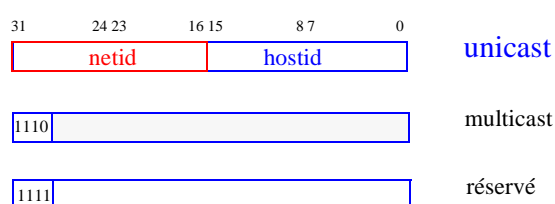
Format de l'adresse IPv4 : 32bits.

Notation décimale pointée ("dotted decimal"). Par exemple 200.1.2.4.

. Adressage hiérarchique :

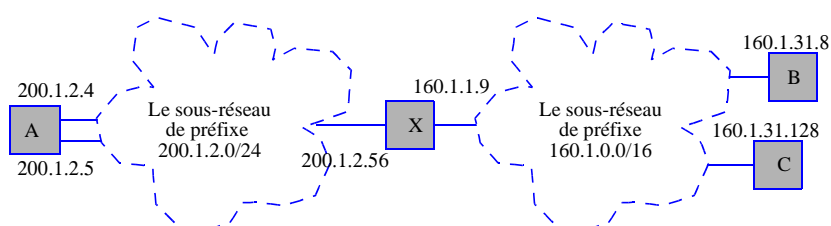
- identificateur du (sous-)réseau (**netid**), appelé aussi "address prefix"
- identificateur de l'interface de la station dans le (sous-)réseau (**hostid**)

. Deux types d'adresse IP : unicast ou multicast



#### 3.2. Utilisation des adresses IP sous Internet

- . Une station IP doit avoir au moins une adresse IP unicast.
- . Une station peut avoir autant d'adresses que d'**interfaces** à un (sous-)réseau  
 ⇒ l'adresse identifie l'interface et pas la station !
- . Toutes les stations connectées à un même (sous-)réseau ont le même préfixe (ont le même **netid**).



### 3.3. Subnet mask

Chaque interface est munie d'un "subnet mask" (rfc 1009).

Le subnet mask permet à une interface d'obtenir son préfixe :

- Exemple : la station d'adresse 160.1.31.8 a pour subnetmask 255. 255. 0.0.
- Son préfixe est :  $160.1.31.8 \& 255.255.0.0 = 160.1.0.0$

Une station source peut déterminer si une station destination est sur le même sous-réseau IP qu'elle-même grâce au masque de sous-réseau.




Exemple :

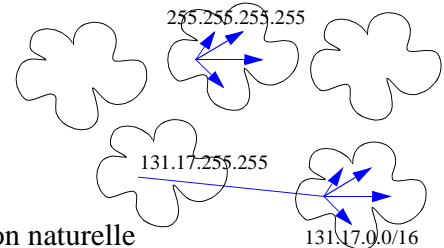
- l'adresse IP de la station source est 160.1.31.8, son masque de sous-réseau est 255.255.0.0, le paquet à envoyer est adressé à la station 160.1.31.128. La station source doit-elle transmettre ce paquet sur le sous-réseau local, ou bien le transmettre à un routeur ?
- Autrement dit :
  - .  $(160.1.31.8 \& 255.255.0.0)$  est-il égale à  $(160.1.31.128 \& 255.255.0.0)$  ?

Sous Unix : commande "`/usr/etc/ifconfig -a`", fichier "`/etc/netmasks`"


### 3.4. Quelques adresses IP particulières

Diffusion : "11...11<sub>2</sub>"

- . diffusion vers toutes les stations du sous-réseau local : 255.255.255.255
- . diffusion vers toutes les stations du sous-réseau distant : D1.D1'.D1".255 ou D2.D2'.255.255 ou D3.255.255.255
- .  pas d'utilisation en tant qu'adresse source
- .  très coûteux si le sous-réseau n'offre pas de diffusion naturelle
- .  utilisé pour l'apprentissage (auto-configuration)

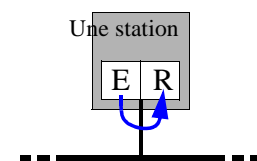


Celle-ci : "00...00<sub>2</sub>"

- . cette station : 0.0.0.0 (quand elle ignore sa propre adresse)
- .  pas d'utilisation en tant qu'adresse de destination

Test : 127

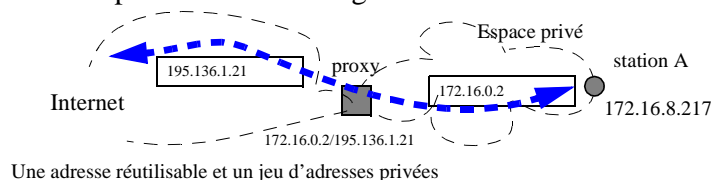
- . écho local des datagrammes (la station reçoit ce qu'elle a émis) : 127.0.0.1 ("loopback")



### 3.5. Adresses IP privées

#### Adresses attribuées et gérées localement (rfc 1918)

- Utilisation exclusivement locale des adresses privées :
  - protection par isolation
- Pas de communications externes, sauf par réutilisation d'adresses générales
  - . un "proxy" chargé de la traduction ("Network Address Translation") entre adresses privées et adresses générales



- Adresses privées de classe A : 10.0.0.0/8
- Adresses privées de classe B : 172.16.0.0/12
  - les  $2^{20}$  adresses de 172.16.0.0 à 172.31.255.255
- Adresses privées de classe C : 192.168.0.0/16

Nota : des stations situées dans des sous-réseaux différents peuvent avoir la même adresse privée

### 3.6. L'agrégation d'adresses

#### Les entrées de la table de routage ayant

- même "next hop" et
- des adresses de destination consécutives (au sens de la représentation binaire)

peuvent être agrégées en une seule entrée.

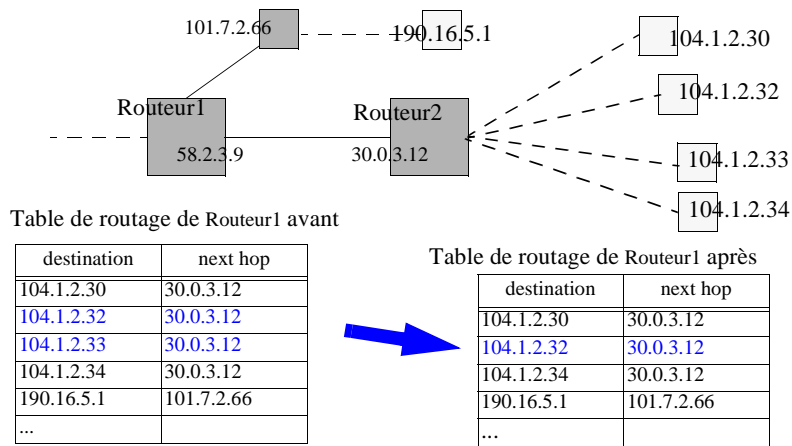
#### Avantages :

- diminue la taille de la table de routage,
- accélère la recherche du prochain routeur

Chaque routeur route un paquet vers le prochain routeur en fonction du "longest prefix match".



## Exemple d'agrégation d'adresses :



## 3.7. Attribution des adresses

Délégation hiérarchique de l'attribution des adresses :

- efficacité

IANA ("Internet Assigned Numbers Authority") attribue des plages d'adresses aux organismes régionaux de régulation des adresses.

Chaque organisme régional attribue une plage d'adresses (parmi celles qu'il gère) aux organismes nationaux.

Chaque organisme national attribue une plage d'adresses (parmi celles qu'il gère) aux organismes locaux qui en ont besoin.

Et ainsi de suite récursivement.

Exemple :

- Europe = RIPE NCC ("Réseaux IP Européens : Network Coordination Center)
- France = AFNIC ("Association Française pour le Nommage Internet en Coopération").
- un fournisseur d'accès à Internet (ISP), une administration ou une entreprise

L'administrateur Internet de l'organisme local :

- baptise chaque station locale
  - . une attribution statique d'une adresse à chaque station
  - . une attribution dynamique (DHCP) : réallocation possible d'une adresse non-utilisée
- à la responsabilité d'organiser ses sous-réseaux IP ("subnet")

## 4. Segmentation des datagrames IP

### 4.1. Présentation

#### Problème :

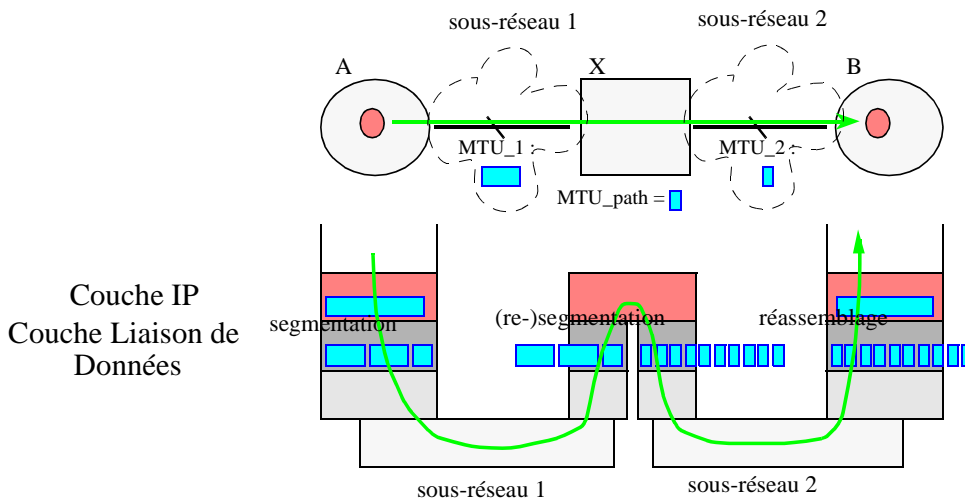
Adaptation de la taille des unités de données à transmettre à la taille des champs de données des trames du réseau local sous-jacent.

- . L'utilisateur veut être capable de transmettre des unités de données de taille quelconque.
- . Les réseaux locaux sous-jacents offrent des champs 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 trames (segmentation).
- . Au récepteur :
  - reconstituer l'unité de données initiale (réassemblage).
- . Aux routeurs :
  - si les réseaux locaux successifs comportent des trames de tailles inférieures : (re-)segmentation.

## 4.2. Exemple

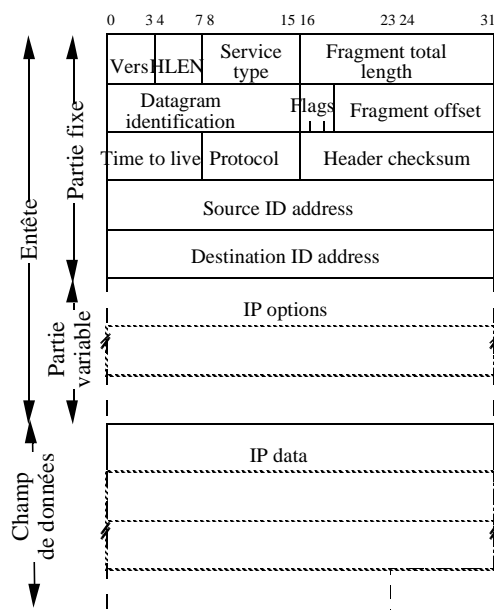


**MTU** : "Maximum Transfer Unit"

- . longueur maximum d'un datagramme IP pour qu'il puisse être transmis sans segmentation sur la liaison du réseau local sous-jacent
- . 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

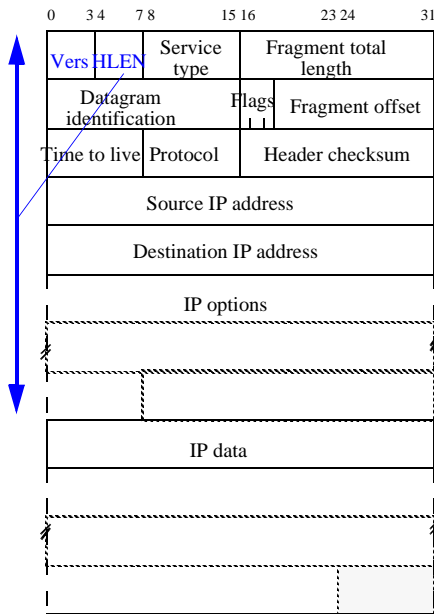
Structure d'un datagramme :

- 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.

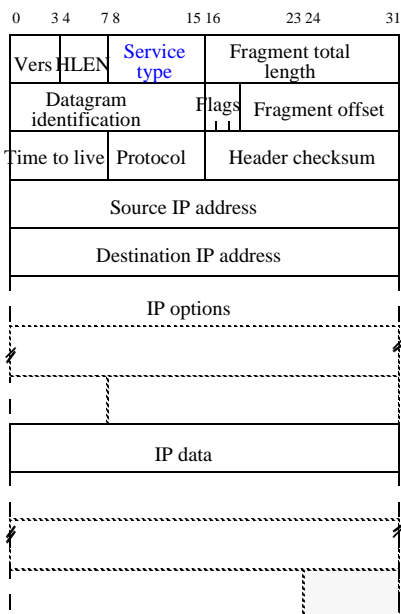
- . Version courante : 4 (septembre 1981)
- . Version nouvelle IPv6 : 6! (décembre 95)
- . Un même réseau peut accueillir différentes versions de protocole : les drivers IP écartent les datagrammes qu'ils ne peuvent pas décoder.

**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

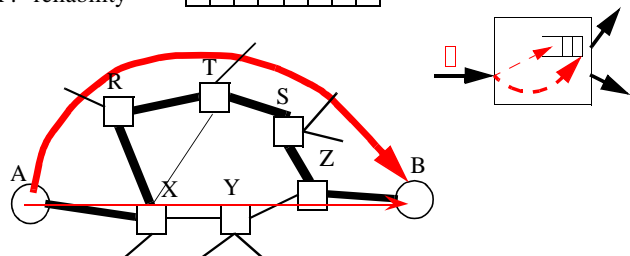
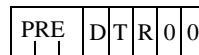


**Service type** (8 bits) : ToS "Type of service"

- . 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"



### 5.3.2 Le champ "Differentiated Services"

Redéfinition du champ ToS (compatible) : rfc 2474, 2598, 2594.



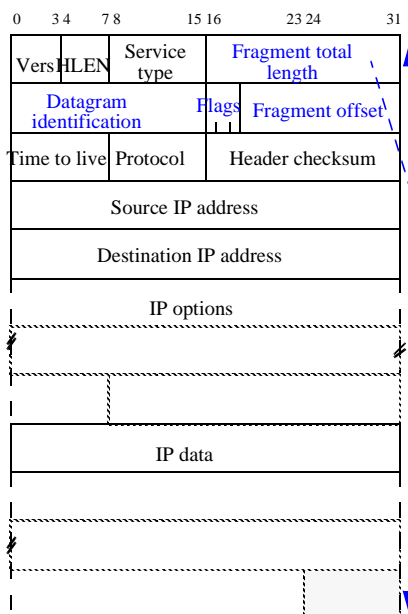
champ Differentiated Services Codepoint

Enrichissement du service "best effort" en introduisant différents types de flux de données qui se verront offrir différents types de services de transmission :

- AF : "Assured Forwarding"
  - . 4 classes se voyant attribuer chacune une certaine portion de la bande passante
    - "gold", "silver", "bronze", and ... "best-effort"
  - . 3 niveaux de priorité de perte au sein de chaque classe
  - . AF1,1: 00101000, AF1,2: 00110000, AF1,3: 00111000, AF2,1: 01001000, etc.
- EF : "Expedited Forwarding"
  - . faible délai, faible niveau de perte
  - . EF : 10111000
- autres configurations pour compatibilité ou pour expérimentation

### 5.4. La fragmentation

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

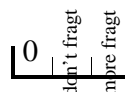


**Fragment total length** (16 bits) : longueur totale du fragment (et pas du datagramme !) en octets (< 64 koctet).

**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.

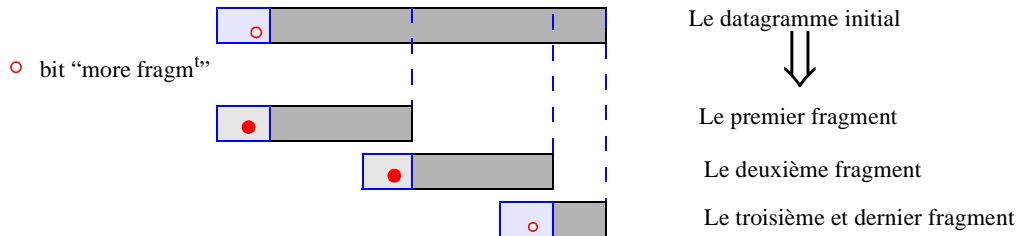


**Fragment offset** (13 bits) : déplacement relatif des fragments par rapport au début du datagramme (en unité de 8 octets).

### 5.4.2 Le traitement du datagramme

#### Constitution des fragments :

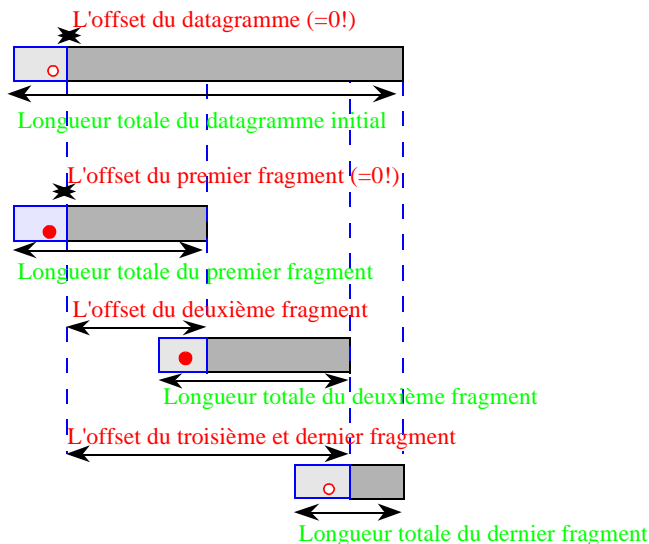
- . Les entêtes des fragments comportent les mêmes champs et ont les mêmes contenus que ceux du datagramme dont ils proviennent :
  - même identification de datagramme, même type de service, même TTL, même protocole, mêmes adresses.
  - sauf pour les champs de longueur, le bit “More fragment” et le “Header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



#### Réassemblage des fragments :

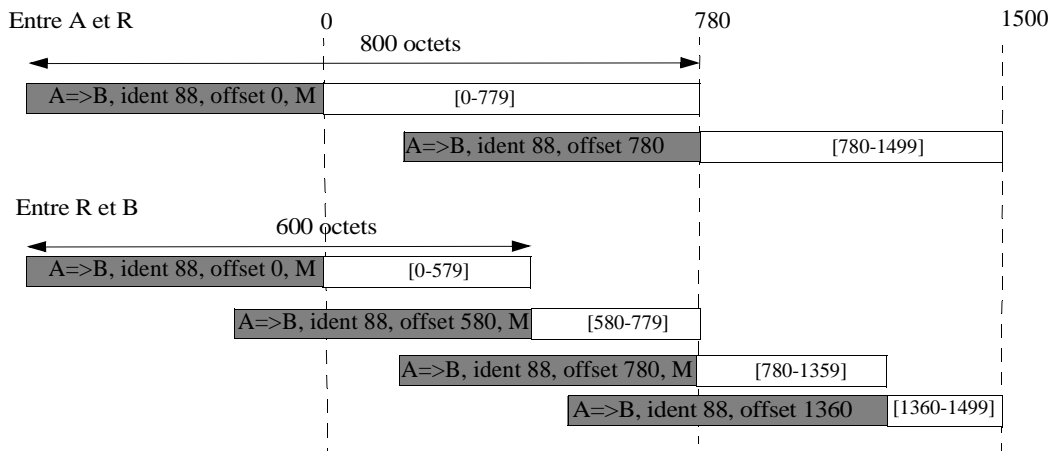
- . Calcul de la taille totale du champ de données du datagramme :
  - offset(**du dernier fragment**) \* 8 + longueur\_totale (**du dernier fragment**)
  - longueur\_de\_l\_entête(**du dernier fragment**) \* 4.
- . Calcul et vérification que tous les fragments sont présents.
- . Reconstitution du champ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

### 5.4.3 Offset et longueur

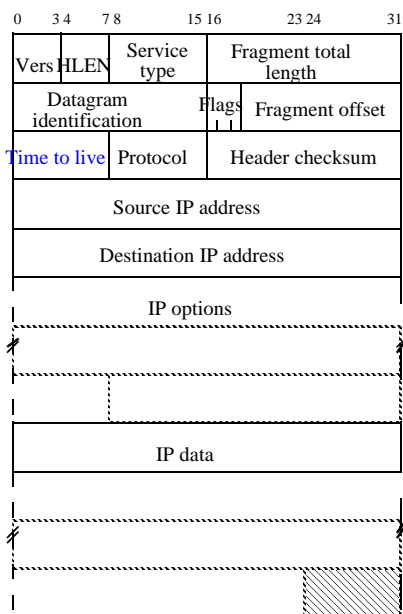


### 5.4.4 Exemple de fragmentation

Fragmentations lors du transport de 1500 octets utiles émis par A vers la destination de B, en supposant que la route passe par un routeur R intermédiaire et que MTU du lien entre A et R est de 800 octets, puis entre R et B le MTU du lien est de 600 octets:



### 5.5. Le champ TTL



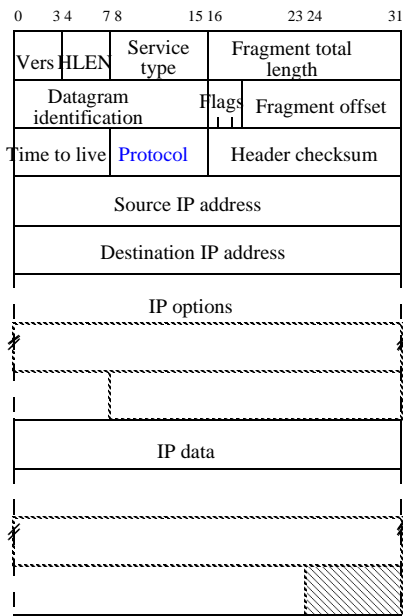
**Time to live** (8 bits) : durée de vie résiduelle du fragment (en nombre de "hops" ou de secondes de résidence).

- . 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.

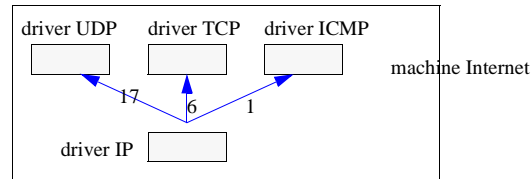
### 5.6. Le champ Protocol du datagramme IP



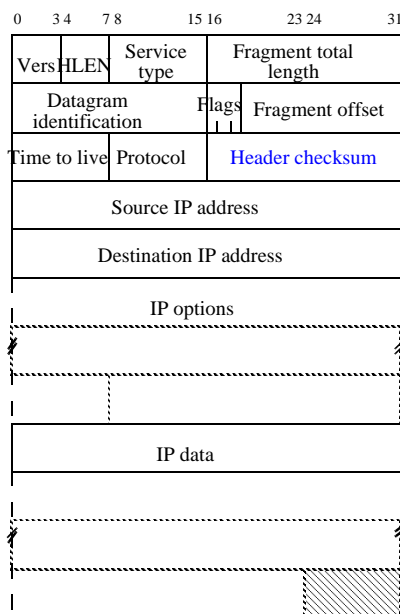
**Protocol (8 bits) :** identification du protocole chargé d'exploiter (décoder) le champ de données.

- . 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 mécanisme de récupération d'erreur sous IP !

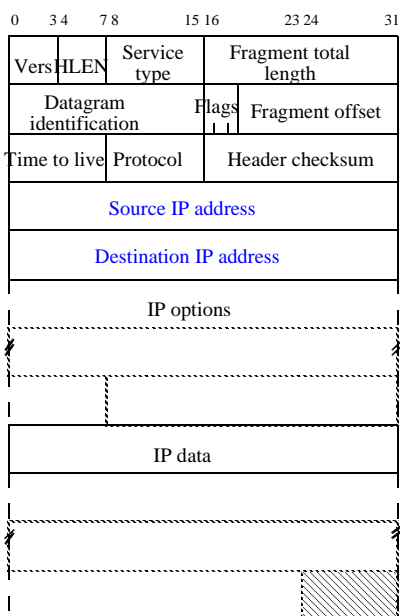
Procédure :

- 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 :  $\text{inv}(4500+05dc+e733+222b+ff11+0000+c02c+4d60+c02c+4d01) = \text{inv}(46e04) = \text{INV}(6e08) = 91f7$



### 5.8. Le champ Address du datagramme IP

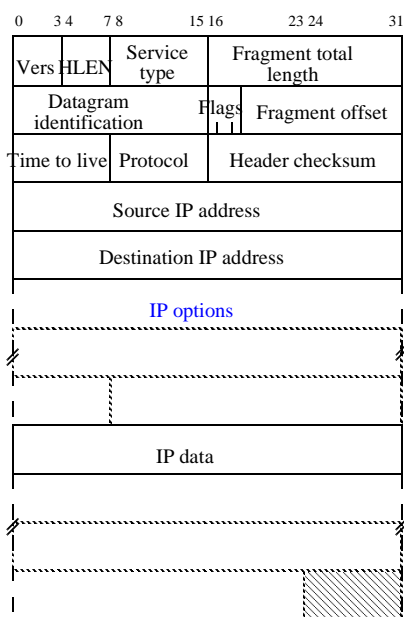


**Source IP address (32 bits) :**  
 identifie la station émettrice.  
 . retour à l'envoyeur (message ICMP).

**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

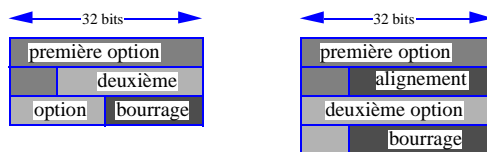
#### 5.9.1 Le champ *IP options* du datagramme IP



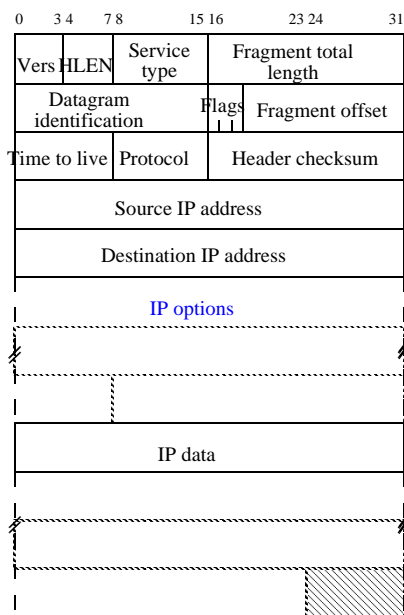
**Partie variable de l'entête**  
 . limitée à 20 octets  
**Leur traitement ralentit le routage :**  
 . certains routeurs ont deux files de datagrammes :  
 - datagrammes avec options  
 - datagrammes sans option  
**Fonction principale :**  
 . choix de la route par l'émetteur

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

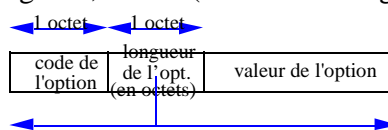
**Exemples de champ *IP options* :**



### 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

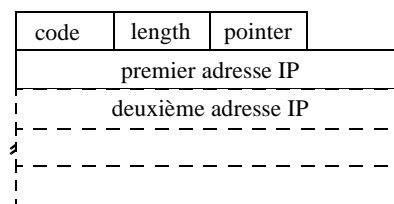
Un chemin = Liste des adresses IP des routeurs  
. 9 au maximum !

Record Route :

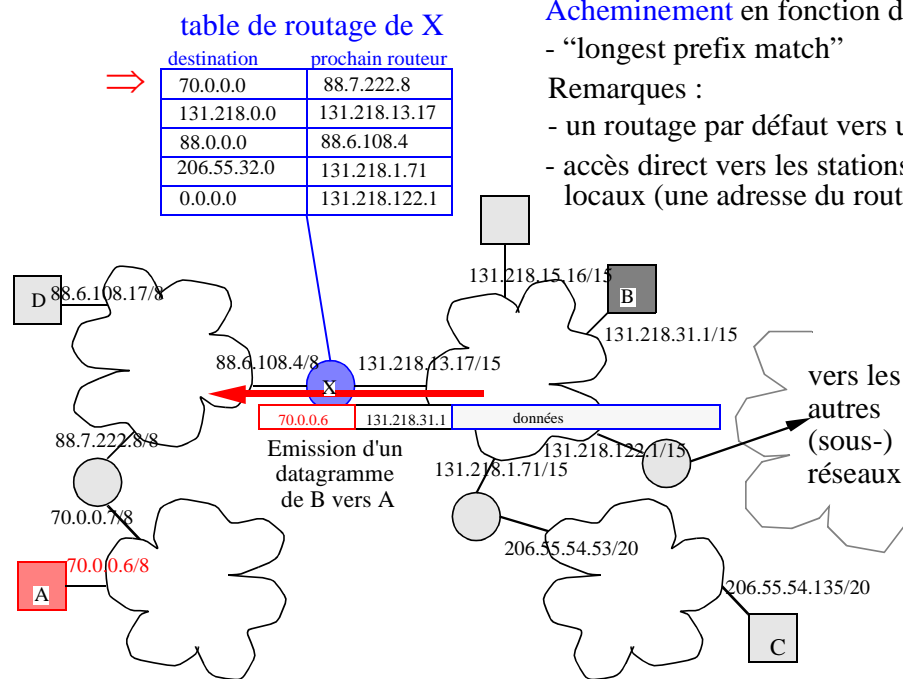
- . l'émetteur prépare une liste vide (pointer=3),
- . qui est remplie par chacun des routeurs du chemin (pointer+=4),
- . tant qu'il y a de la place dans la liste (pointer<length).

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"



## 7. Conclusion

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

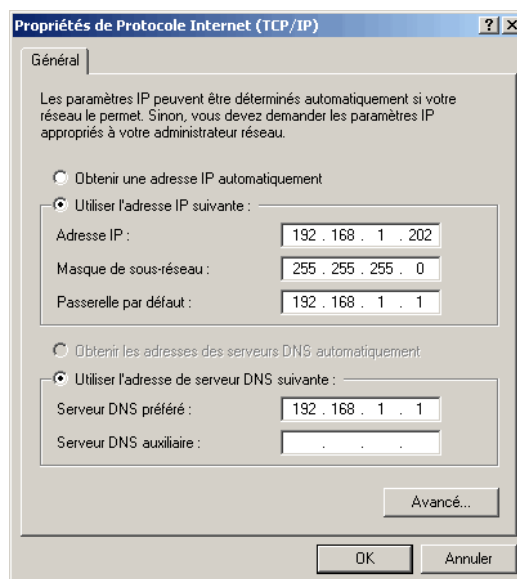
Un datagramme IP constitué des octets suivants représentés en hexadécimal en lisant de gauche à droite et de haut en bas (par exemple : le 1<sup>er</sup> octet vaut  $49_{16}$ , le second :  $00_{16}$  et le cinquième :  $01_{16}$ ).

49	00	00	30
01	23	00	00
0D	11	42	DB
83	FE	3D	0D
C0	2C	4D	51
07	0F	0B	83
FE	0B	01	C0
2C	4D	4D	00
00	00	00	00
84	01	00	07
00	0C	00	00
01	02	03	04

Quelle est la longueur de l'entête de ce datagramme IP ? Quelle est la valeur de l'adresse de destination ? Quelle est la valeur du premier octet du champ de données ? Quelle est la longueur totale de ce fragment ? Quelle est la longueur totale du datagramme (s'il est possible de le calculer à l'aide de ce seul fragment) ?

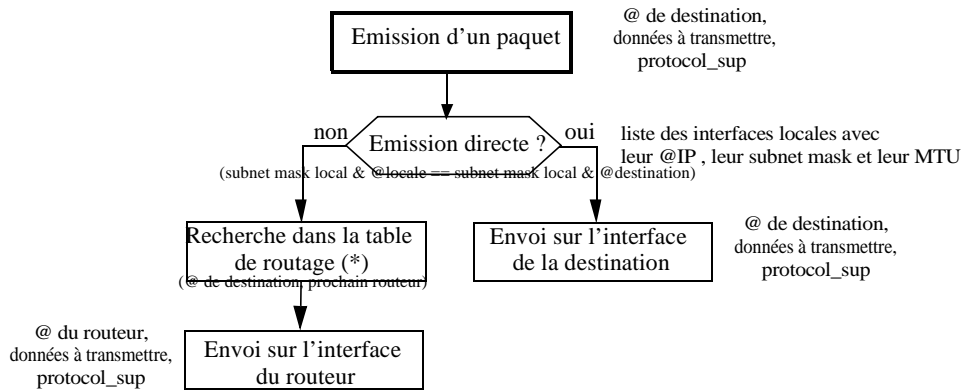
### 7.2. Exemple de configuration d'une station Windows pour Internet

Dans la fenêtre *Panneau de configuration* double-cliquez sur l'icône *Connexions réseau et Internet*. Dans la fenêtre *Propriétés de Connexion au réseau local*, sélectionnez *Protocole Internet (TCP/IP)*, puis cliquez sur le bouton *Propriétés*.



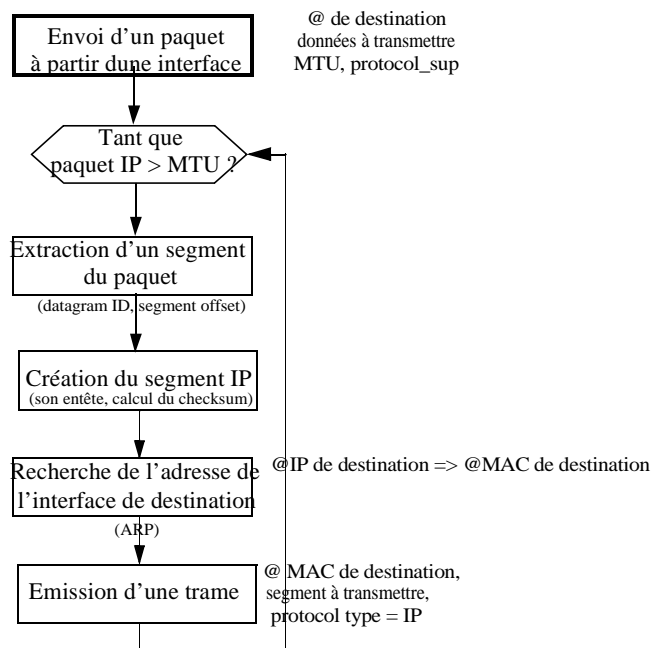
### 7.3. Emission d'un paquet IP

Choix de l'interface à partir duquel le paquet va être émis.



(\*) La table de routage contient au moins une entrée pour le routeur par défaut.

### L'algorithme d'envoi d'un paquet IP à partir d'une interface



### 7.4. L'algorithme de réception d'un segment IP

