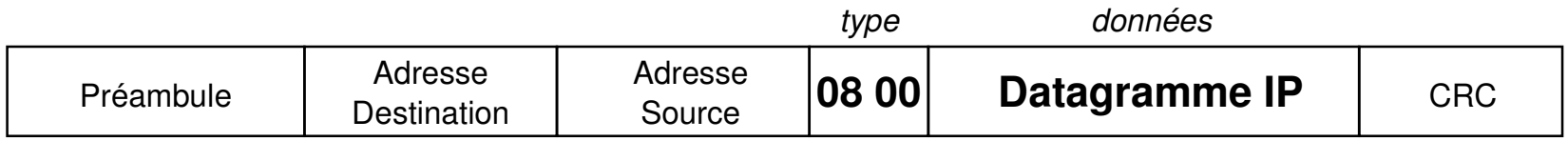
Datagrammes IP

(RFC 791)

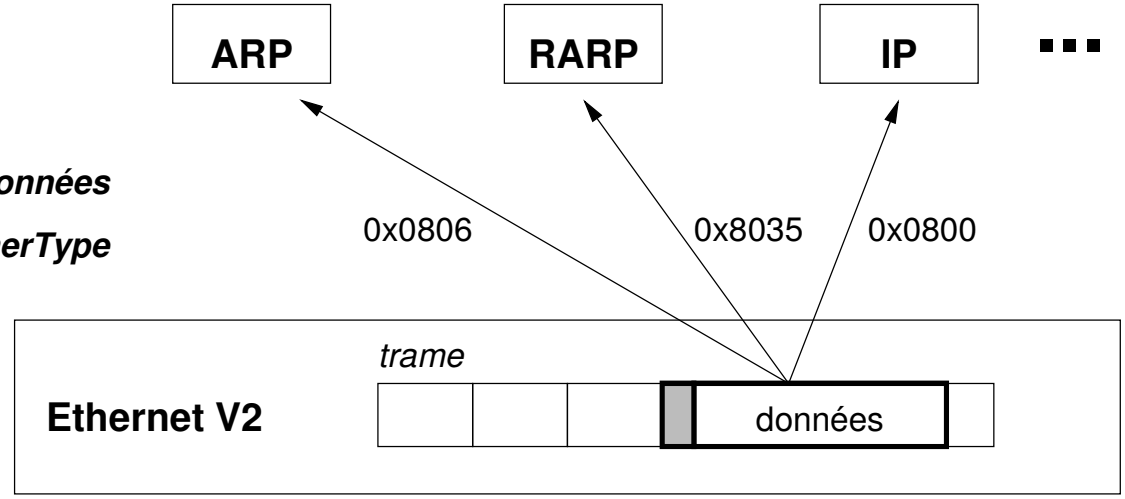
Datagramme IP sur Ethernet V2

- trame Ethernet v2 contenant un datagramme IP (*EtherType* en Hexa) :



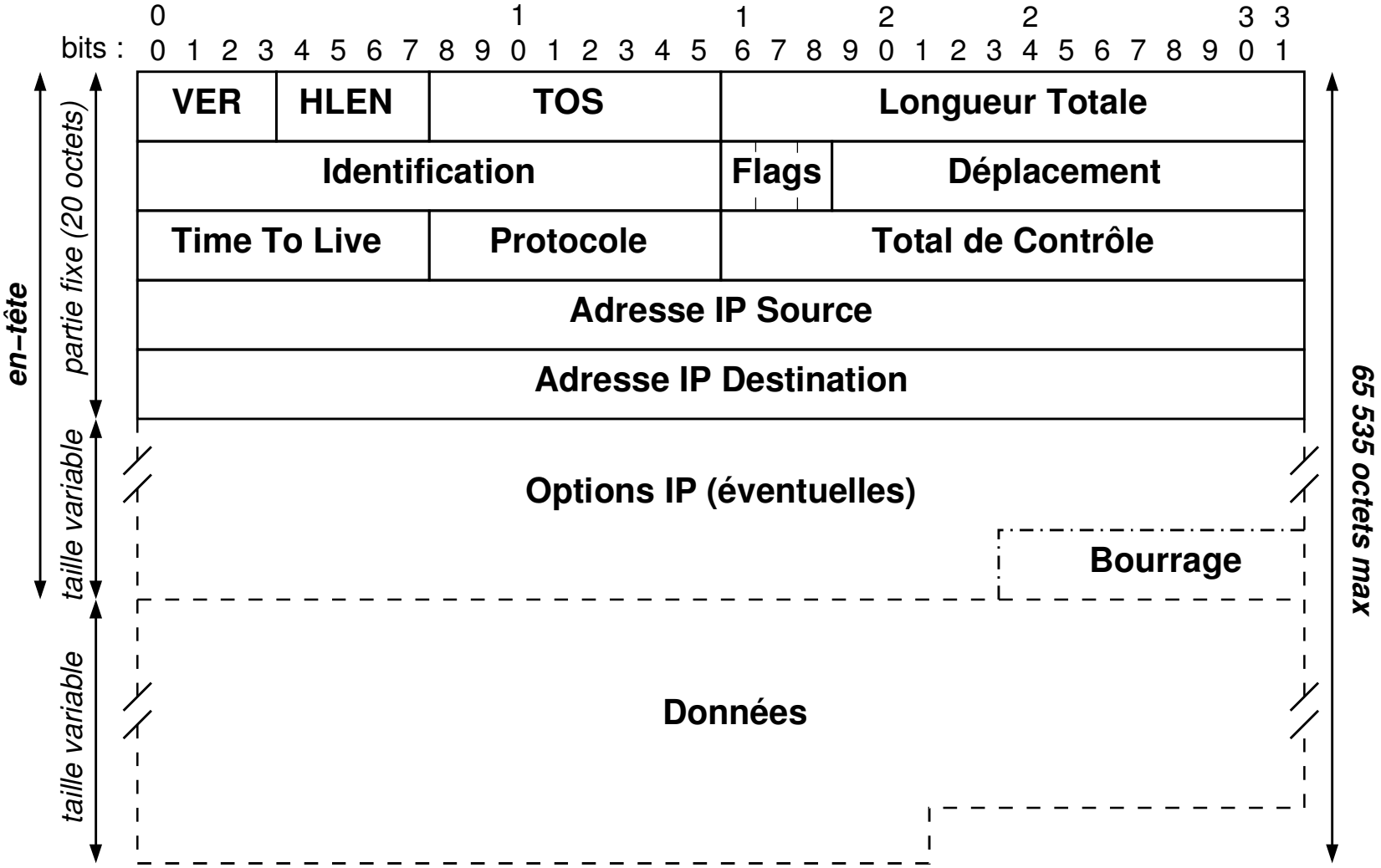
- (dé)multiplexage Ethernet v2 :

le protocole destinataire des données est indiqué par le champ EtherType

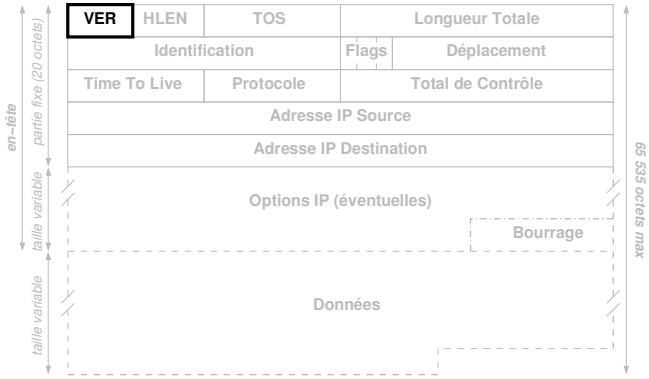


Format du datagramme IP

- en-tête : nombre variable d'octets (multiple de 4)
- données : nombre quelconque d'octets (limité à 65 315)

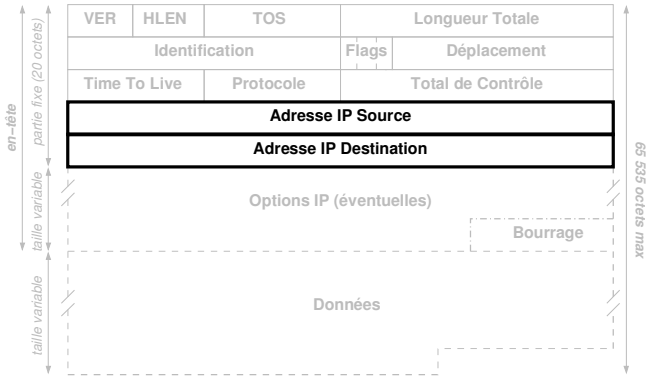


Champ Version

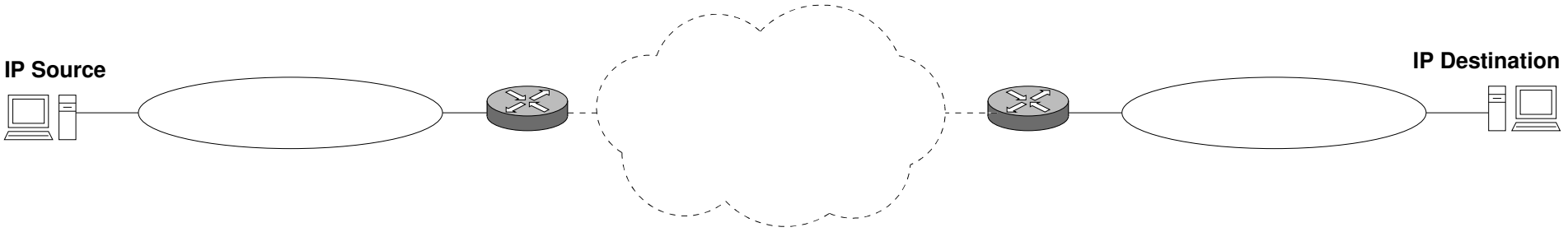


- codé sur 4 bits
- identifie la version du (format du) datagramme
- actuellement, la version est 4 (codée 0100 en binaire)
- dans le datagramme IPv6, ce champ est maintenu et vaut 6
- permet de s'assurer que le datagramme sera correctement interprété

Champs Adresses

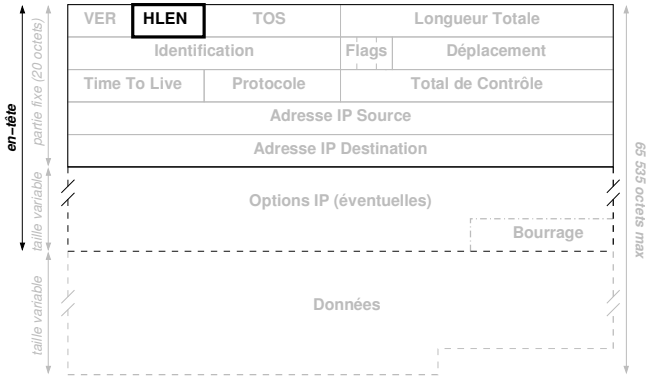


- *adresse IP Source* : (32 bits)
  identifie l'**hôte à l'origine du datagramme**
- *adresse IP Destination* : (32 bits)
  identifie le **destinataire final du datagramme**



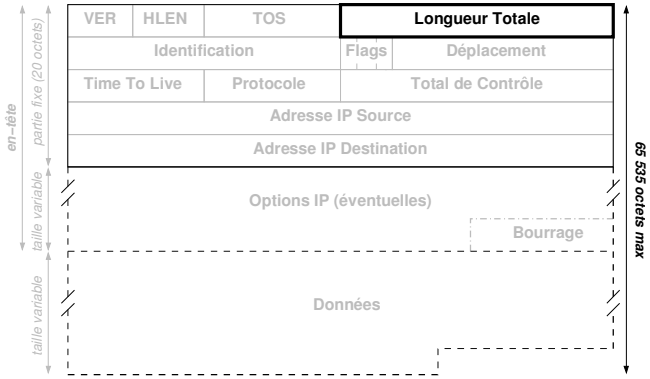
Ces adresses ne sont pas modifiées par les routeurs. Toutefois, en cas de NAT/NAPT (translation d'adresse), la NATbox peut les modifier (voir second semestre).

Champ Longueur d'en-tête (HLEN)



- *(internet) Header LENgth*
- codée sur 4 bits
- indique le nombre de mots de 32 bits de l'en-tête (comprenant les options) :
  - en-tête de 20 à 60 octets
  - $5 \leq HLEN \leq 15$
- si $HLEN > 5$ alors il y a des options

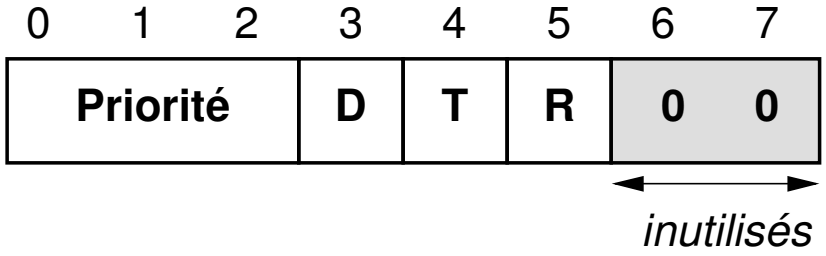
Champ Longueur Totale

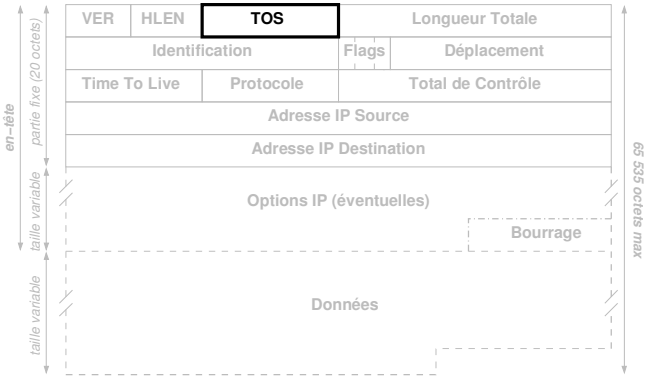


- codée sur 16 bits
- indique le nombre de total d'octets du datagramme (en-tête + données)
- comprise entre 20 et 65 535

Champ Type Of Service (TOS)

- codé sur 8 bits :





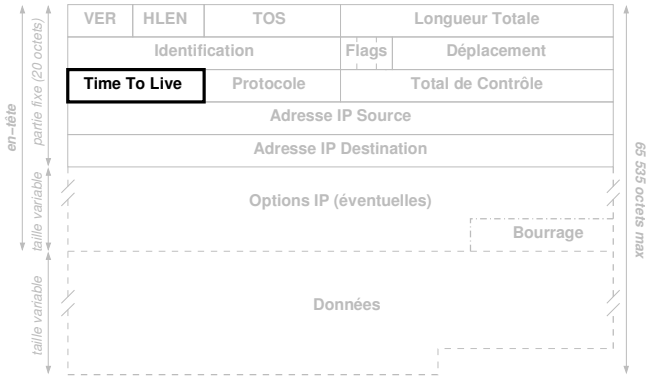
- **Priorité** : de 0 à 7
  - distinction entre "normal" et "controle"
  - routeurs : infos trafic 6 et 7

- bits **D**, **T** et **R** : type d'acheminement **désiré** :
  - **D**(elay) : delai d'acheminement court
  - **T**(hroughput) : débit de transmission élevé
  - **R**(eliability) : grande fiabilité

- le *TOS* constitue un **souhait**, souvent ignoré

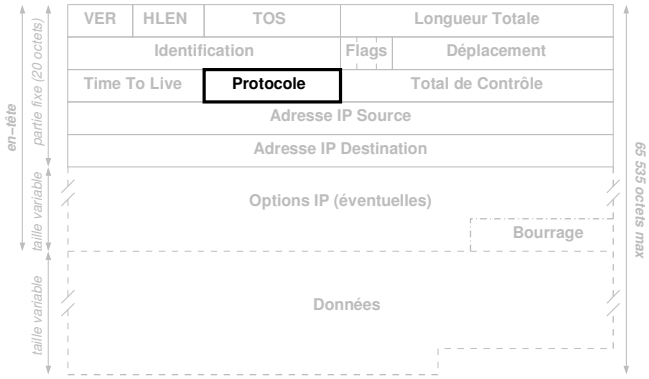
Priorité	
val_2	**signification**
000	routine
001	priority
010	immediate
011	flash
100	flash override
101	critic
110	internetwork control
111	network control

Champ Time To Live (TTL)



- codé sur 8 bits
- indiqué par l'émetteur pour limiter :
  - la "durée de vie" du datagramme (en secondes)
  - le nombre de routeurs traversés par le datagramme
- décrémenté par routeurs et stations traitant le datagramme :
  - de 1 à chaque traversée d'un routeur
  - du temps passé en file d'attente
- si atteint 0, le datagramme est détruit, et l'émetteur est informé par un message ICMP
- évite qu'un datagramme ne circule indéfiniment
- évite que des *fragments* ne soient gardés inutilement

Champ Protocole

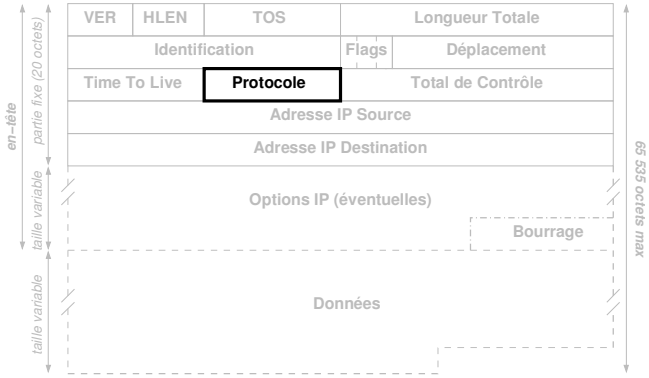


- codé sur 8 bits
- indique le protocole devant recevoir les données du datagramme
- les valeurs de ce champ sont gérées par l'autorité centrale et accessibles sur le site `www.iana.org`

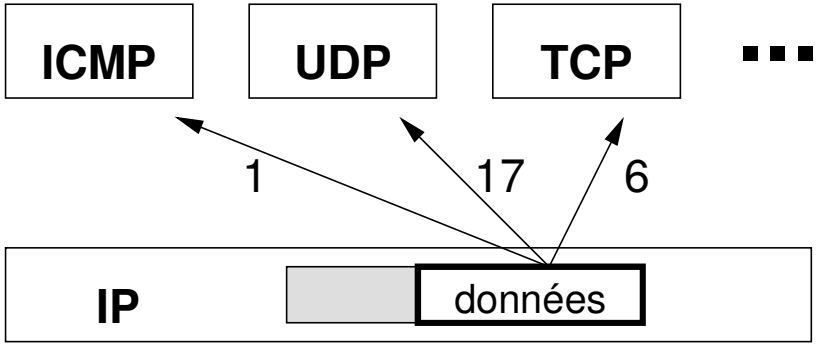
Champ Protocole

- quelques valeurs officielles :

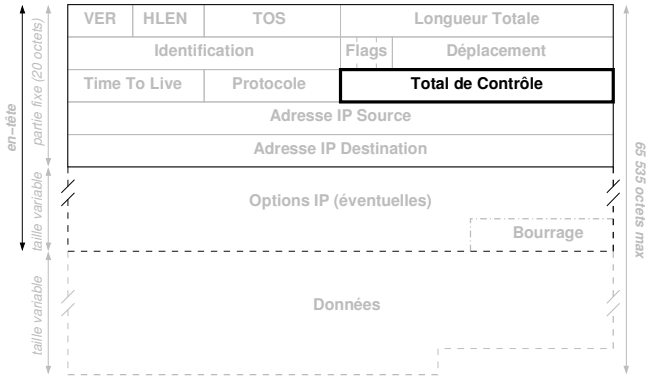
$\mathbf{val}_{10}$	**protocole**
0	IP
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF



- démultiplexage IP :



Champ Total de Contrôle d'en-tête (checksum)

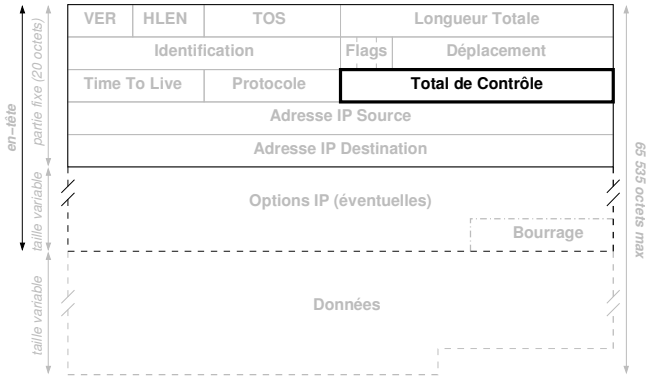


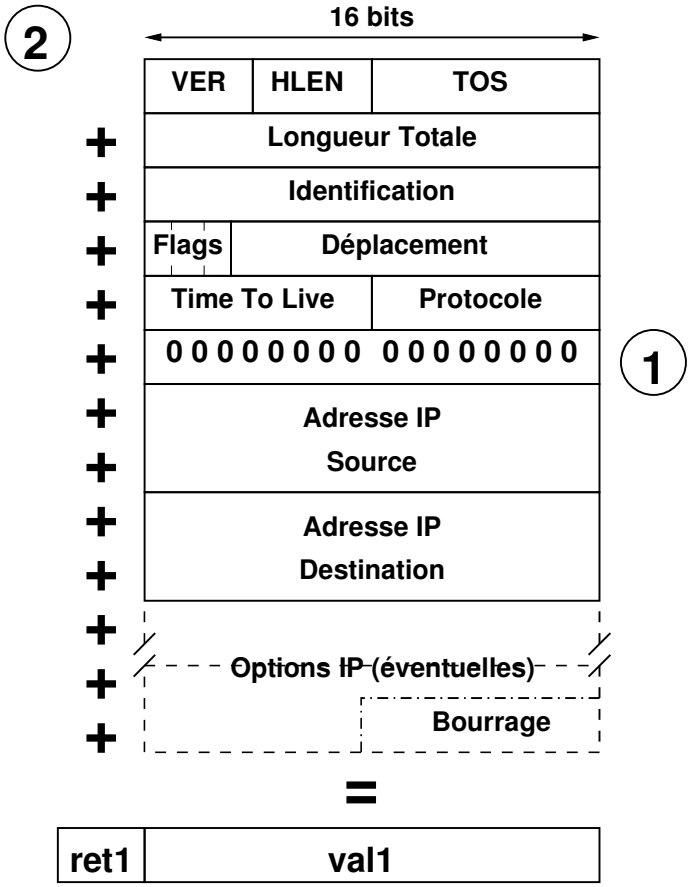
- codé sur 16 bits
- contrôle l'intégrité de l'en-tête **uniquement**

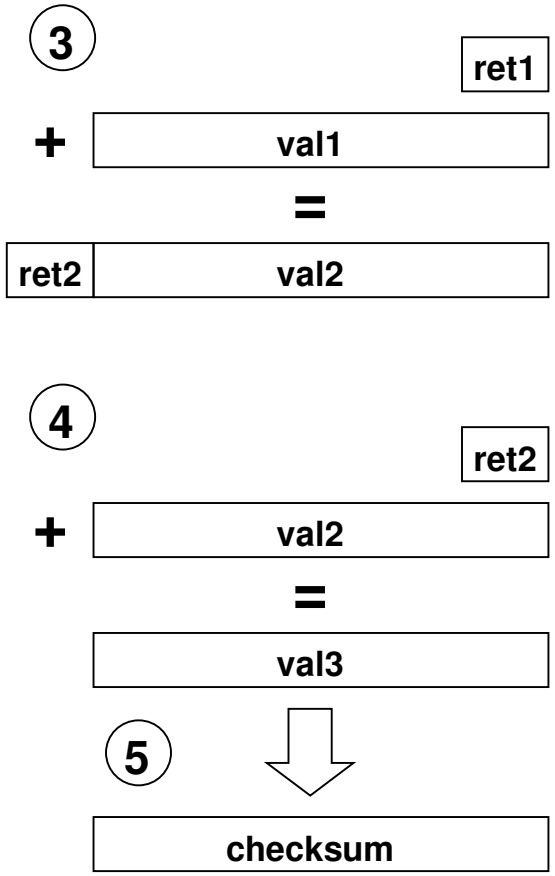
> **IP ne vérifie pas si les données ont subi des erreurs de transmission**

- calculé par l'émetteur
- vérifié lors dela réception (routeurs et destinataire) :
  - stocker le *checksum*
  - calculer le *checksum*
  - si différents alors détruire le datagramme
- recalculé et modifié par les routeurs (car modifient au moins le TTL)

Champ Total de Contrôle d'en-tête : calcul



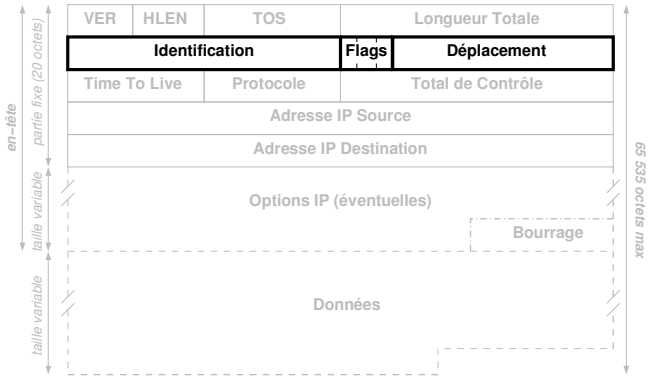




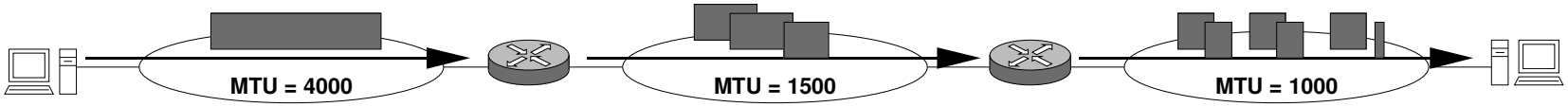
1. mettre checksum à 0
2. calculer somme des mots de 16 bits de l'en–tête
3. ajouter ret1 à val1
4. ajouter ret2 à val2
5. checksum = complément à 1 de val3

MTU et fragmentation

- Maximum Transfer Unit (MTU) :
  - taille max des données (charge utile) transportées sur un réseau physique
    - Ethernet : 1 500 octets
    - Token Ring : 4 ou 16 Ko
    - X.25 : 128 octets recommandés (max 255)
    - SMDS : 9 188 octets
    - Frame Relay : 1 600 octets
  - . . . et donc des datagrammes

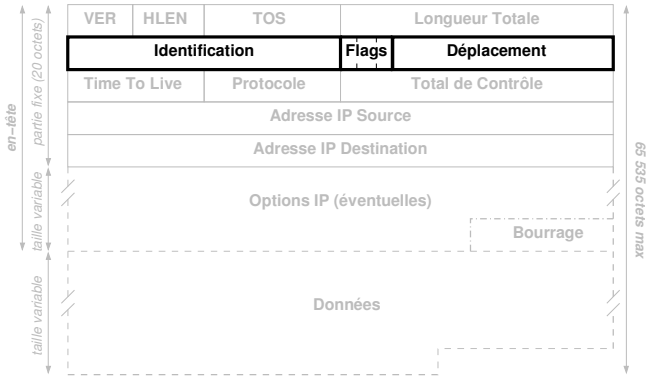


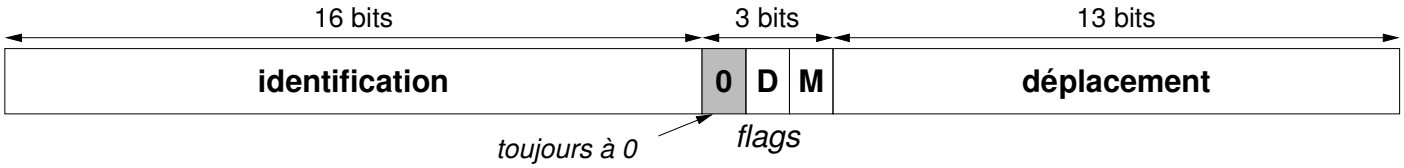
- IP **fragmente** tout datagramme plus grand que le MTU du réseau qui doit le transporter :



- chaque **fragment est un datagramme** acheminé indépendamment (peut suivre une route différente des autres fragments) et peut être à son tour fragmenté

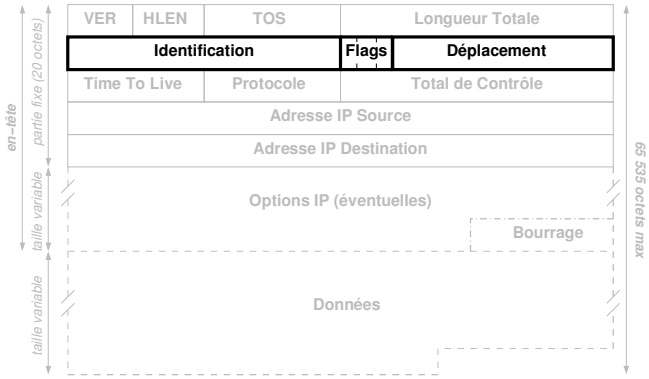
Champs Fragmentation





- *identification* : valeur identifiant le datagramme d'origine relativement à l'adresse IP Source
- bit *D(on't Fragment)* : le datagramme ne doit pas être fragmenté (détruit et message ICMP si impossible)
- bit *M(ore)* : à 0 si ce datagramme est le dernier (ou seul) fragment
- *déplacement* (*Offset*) : indique la position du premier octet de données dans le datagramme d'origine. Cette position est *déplacement* $\times 8$. Vaut 0 si pas de fragmentation

Réassemblage

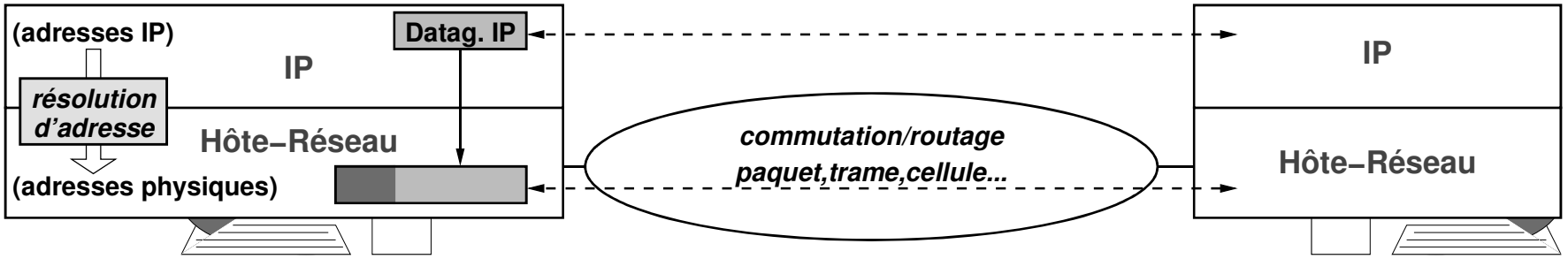


- réalisé par le destinataire final :
  - met en attente les fragments des datagrammes incomplets
  - les réordonne
  - détruit tous les fragments d'un datagramme si le TTL de l'un d'eux passe à 0 (et envoie un message ICMP à l'émetteur)

Résolution d'adresse

Nécessité de la résolution d'adresse

situation : une station/routeur S a un datagramme à transmettre à une station/routeur D du même réseau. D est l'adresse IP de la destination finale du datagramme (remise directe) ou celle d'un routeur obtenue par consultation de la table de routage (remise indirecte)



- la transmission doit se faire en utilisant le service de la couche hôte-réseau (réseau physique) (couche liaison modèle OSI)
- la couche hôte-réseau n'utilise pas les adresses IP mais des adresses physiques (adresses MAC)
- la **résolution d'adresse** est le mécanisme permettant d'obtenir l'adresse physique (de l'interface/carte réseau) de la station possédant une certaine adresse IP

Méthodes possibles pour la résolution d'adresse

résolution directe

- l'adresse physique est déterminée comme une fonction de l'adresse IP
- méthode simple à mettre en œuvre si les adresses physiques sont configurables

interrogation d'un serveur

- un serveur est chargé de collecter les adresses physiques et IP des hôtes du réseau
- les stations interrogent le serveur pour résoudre les adresses
- méthode souvent utilisée lorsque le réseau ne permet pas la diffusion
- mais la résolution n'est plus possible si le serveur devient injoignable...

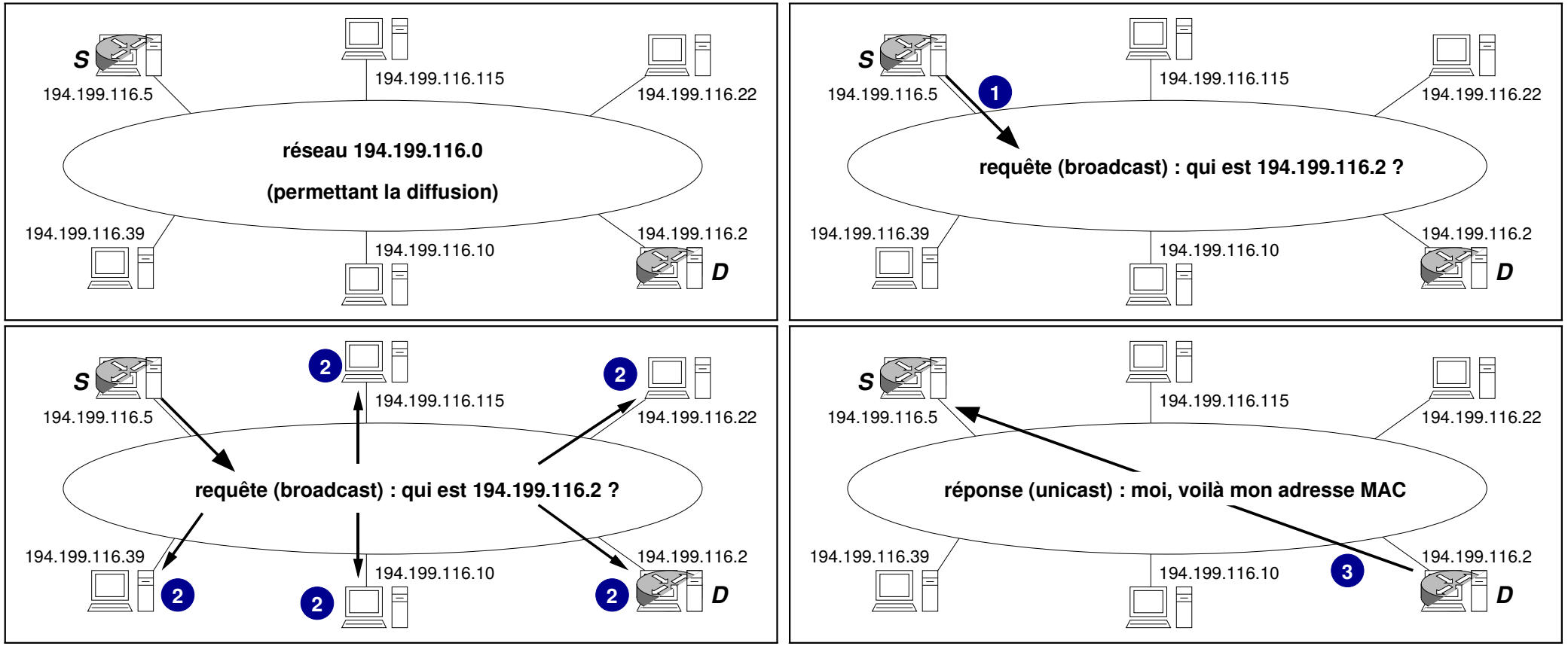
Résolution dynamique par ARP

Pour les réseaux permettant la diffusion, la méthode privilégiée est ARP (*Address Resolution Protocol*), définie dans la RFC 826.

- ARP a l'avantage d'être à la fois dynamique et décentralisée :
  - les changements d'association adresse IP/adresse MAC sont automatiquement et rapidement pris en compte
  - aucun serveur n'est nécessaire et une panne d'une station n'a aucun impact global
- ARP a été originellement défini pour IP et Ethernet. Mais il est plus général et peut être utilisé sur tout type de réseau permettant la diffusion, pour le compte de différents protocoles réseau (dont IP)

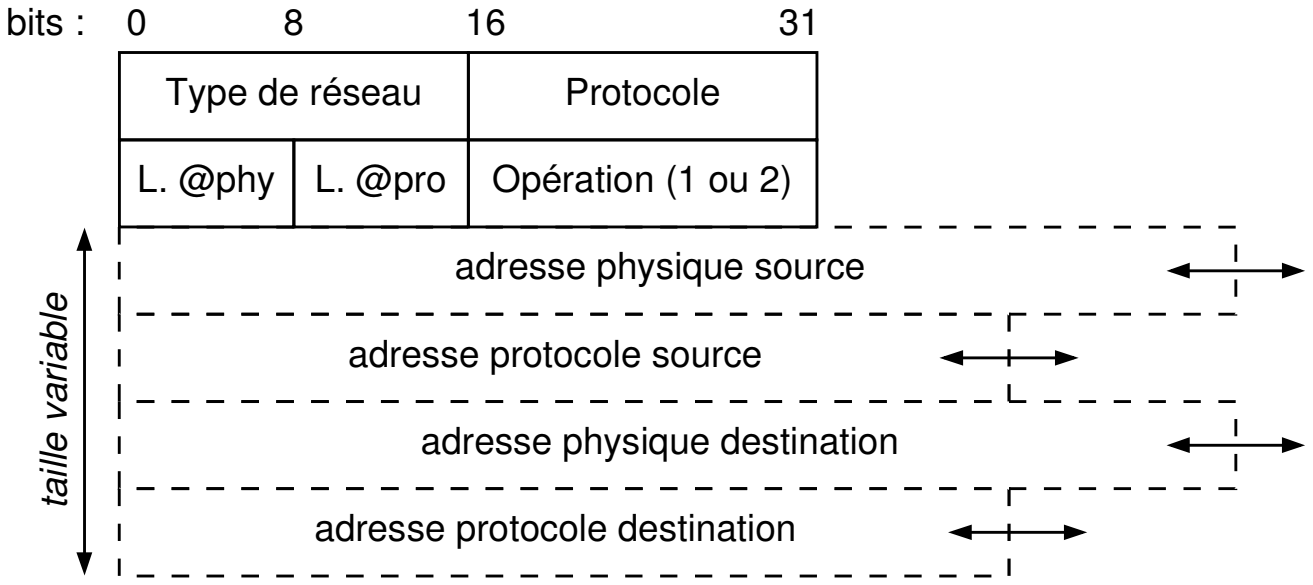
IP sur Ethernet utilise systématiquement ARP

Principe de la résolution par ARP



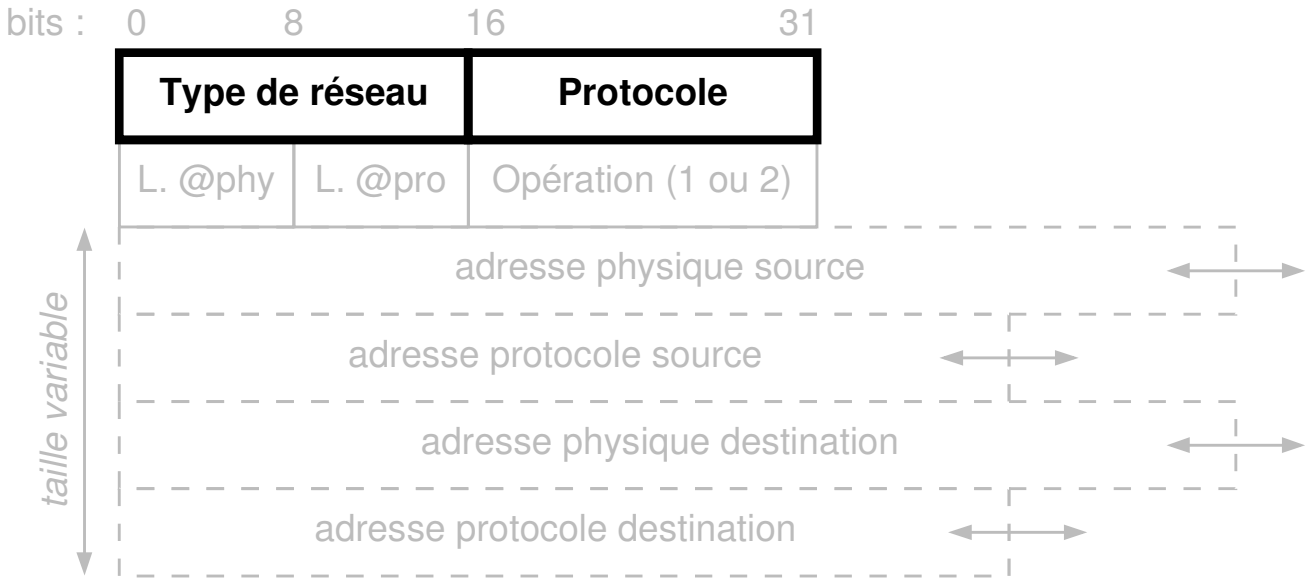
1. S envoie en **broadcast** une **requête ARP** signifiant qu'il souhaite obtenir l'adresse physique correspondant à D
2. la requête est reçue et traitée par toutes les stations du réseau
3. seule la station d'adresse D répond en envoyant en **unicast** à S une **réponse ARP** contenant l'adresse physique demandée

Format du datagramme ARP



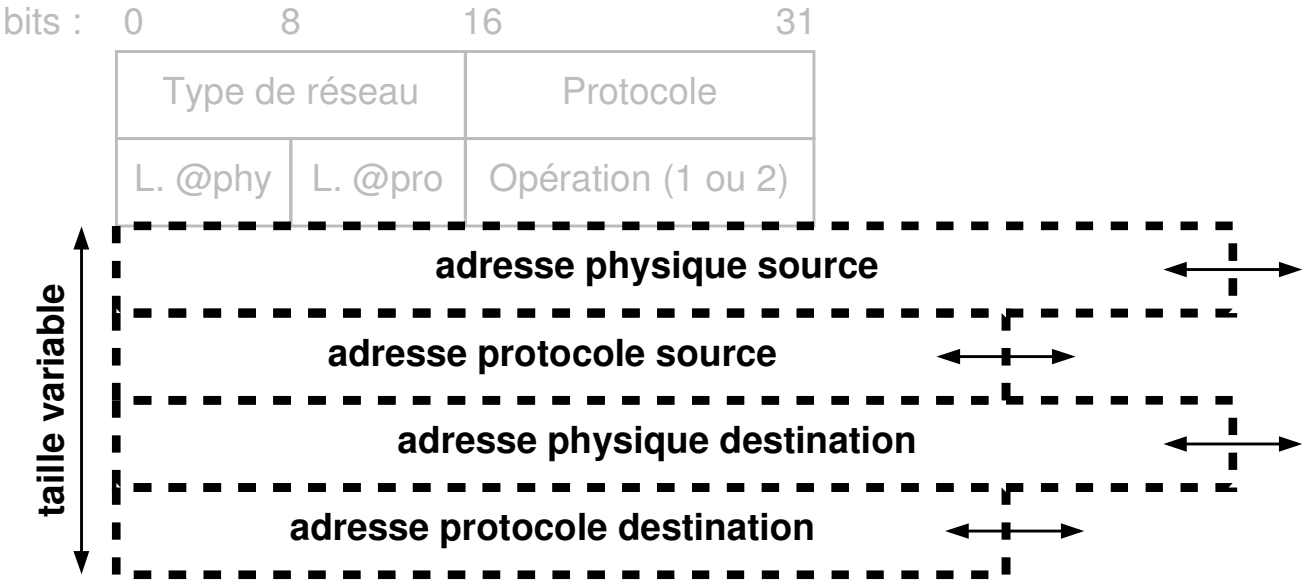
- la taille du datagramme ARP dépend des protocoles en jeu
  - la taille (en octets) des adresses physiques (comme Ethernet) est indiquée par le champ *Longueur adresses physiques* (L. @phy)
  - la taille (en octets) des adresses protocole (comme IP) est indiquée par le champ *Longueur adresses protocole* (L. @pro)
- les requêtes et les réponses ont le même format ; le champ *Opération* indique s'il s'agit d'une requête (*Opération* vaut 1) ou d'une réponse (*Opération* vaut 2)

Datagramme ARP : type de réseau et protocole



- *Type de réseau* précise le réseau physique utilisé et donc le type d'adresse recherchée
- *Protocole* précise la couche réseau utilisée et donc le type d'adresse à partir duquel la résolution doit être opérée
- les valeurs que peuvent prendre ces champs sont définies par l'IANA (`www.iana.org`) :
  - *Type de réseau* vaut 0x0001 pour Ethernet
  - *Protocole* vaut 0x0800 pour IP

Datagramme ARP : adresses



- qu'il s'agisse d'une requête ou d'une réponse :
  - *adresse physique source* contient l'adresse physique de l'émetteur du datagramme
  - *adresse protocole source* contient son adresse réseau
- *adresse physique destination* est inconnue pour une requête (00:00:00:00:00:00 pour Ethernet), et celle du destinataire pour une réponse
- *adresse protocole destination* contient l'adresse réseau du destinataire (dans une requête, c'est l'adresse à résoudre)

Requête ARP pour IP sur Ethernet V2

adresses de l'émetteur

IP : 194.199.116.5

ethernet : 08:00:05:0e:ab:51

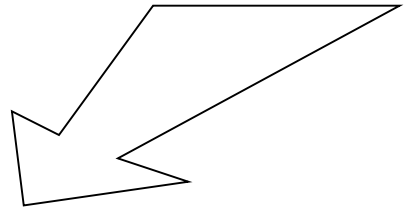
adresses de la cible

IP : 194.199.116.2

ethernet (recherchée) : 08:00:07:5c:10:0a

- Trame Ethernet V2 (en hexadécimal) :

	destination	*source*	*type*	*données*	
préambule	ff:ff:ff:ff:ff:ff	08:00:05:0e:ab:51	**08 06**	**Datagramme ARP (requête)**	CRC
	(broadcast)				



- Requête ARP (en binaire) :

Type réseau : Ethernet (1)	Protocole : IP (0x0800)	L. @ phy : 6	L. @ pro : 4	Opération : requête (1)	
0000000000000001	0000100000000000	00000110	00000100	0000000000000001	
000010000000000000001010000111010101011010010001					← Ethernet source (08:00:05:0e:ab:51)
11000010110001110111010000000101					← IP source (194.199.116.5)
000000000000000000000000000000000000000000000000					← Ethernet destination (inconnue)
11000010110001110111010000000010					← IP destination (194.199.116.2)

Réponse ARP pour IP sur Ethernet v2

émetteur de la réponse

IP : 194.199.116.2

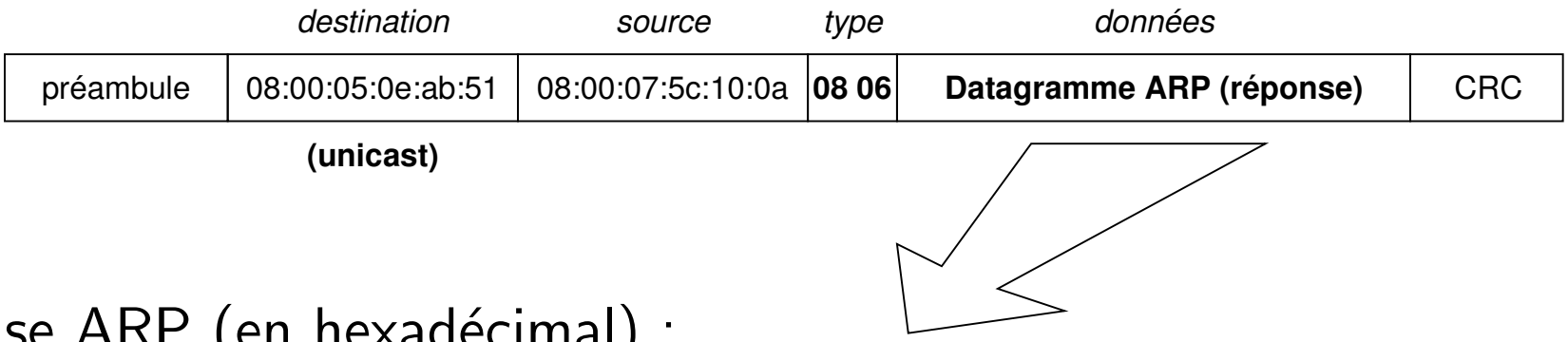
ethernet : 08:00:07:5c:10:0a

destinataire de la réponse

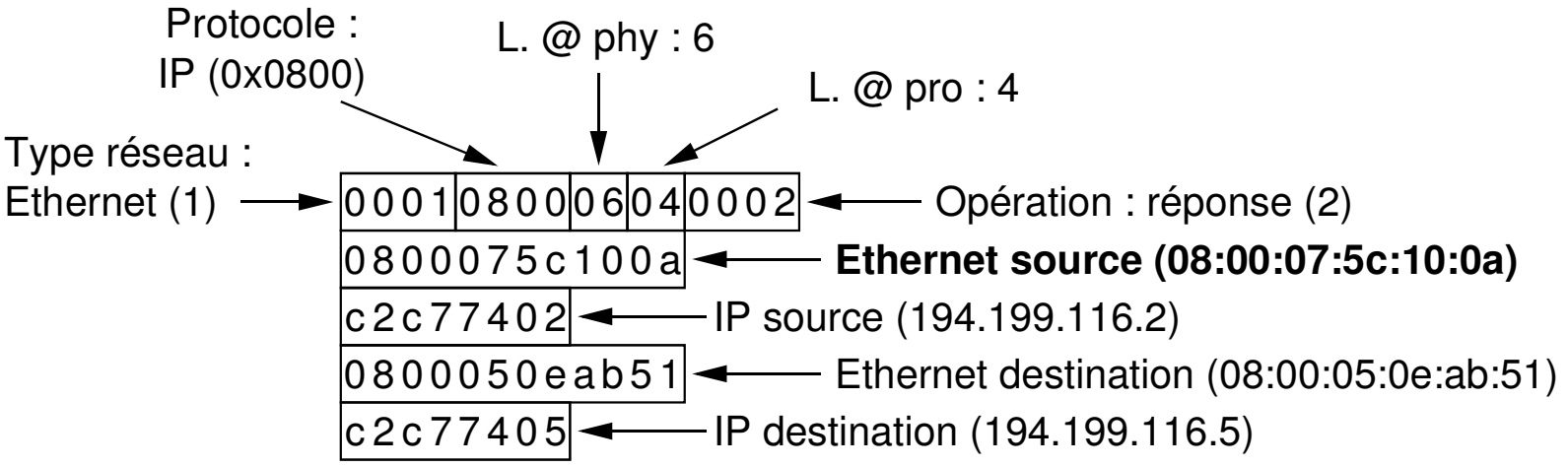
IP : 194.199.116.5

ethernet : 08:00:05:0e:ab:51

- Trame Ethernet V2 (en hexadécimal) :



- Réponse ARP (en hexadécimal) :



Optimisations d'ARP

- **cache** (mémoire temporaire) ARP obligatoire stocké sur les hôtes :
  - contient une liste d'associations $\prec$ adresse MAC, adresse IP $\succ$
  - évite d'émettre une nouvelle requête lorsque l'association a déjà été obtenue
  - une association a une durée de vie limitée (environ 20 minutes)
  - chaque fois qu'une association est confirmée, sa durée de vie est remise à 20 min
  - les associations dont la durée de vie expire sont supprimées
- traitement de la **requête** :
  - les requêtes étant envoyées en broadcast, toutes les stations les traitent
  - or elles incluent l'adresse MAC et l'adresse IP de l'émetteur
  - en recevant une requête, les stations mettent à jour leur cache avec les infos sur l'émetteur
- émission d'une **requête ARP fictive** si changement de carte (et donc d'adresse MAC) :
  - en plaçant sa propre adresse IP comme celle recherchée
  - personne ne répondra mais tout le monde aura mis à jour son cache avec la nouvelle adresse MAC