

V - Couche Réseau : IP, ICMP, ARP

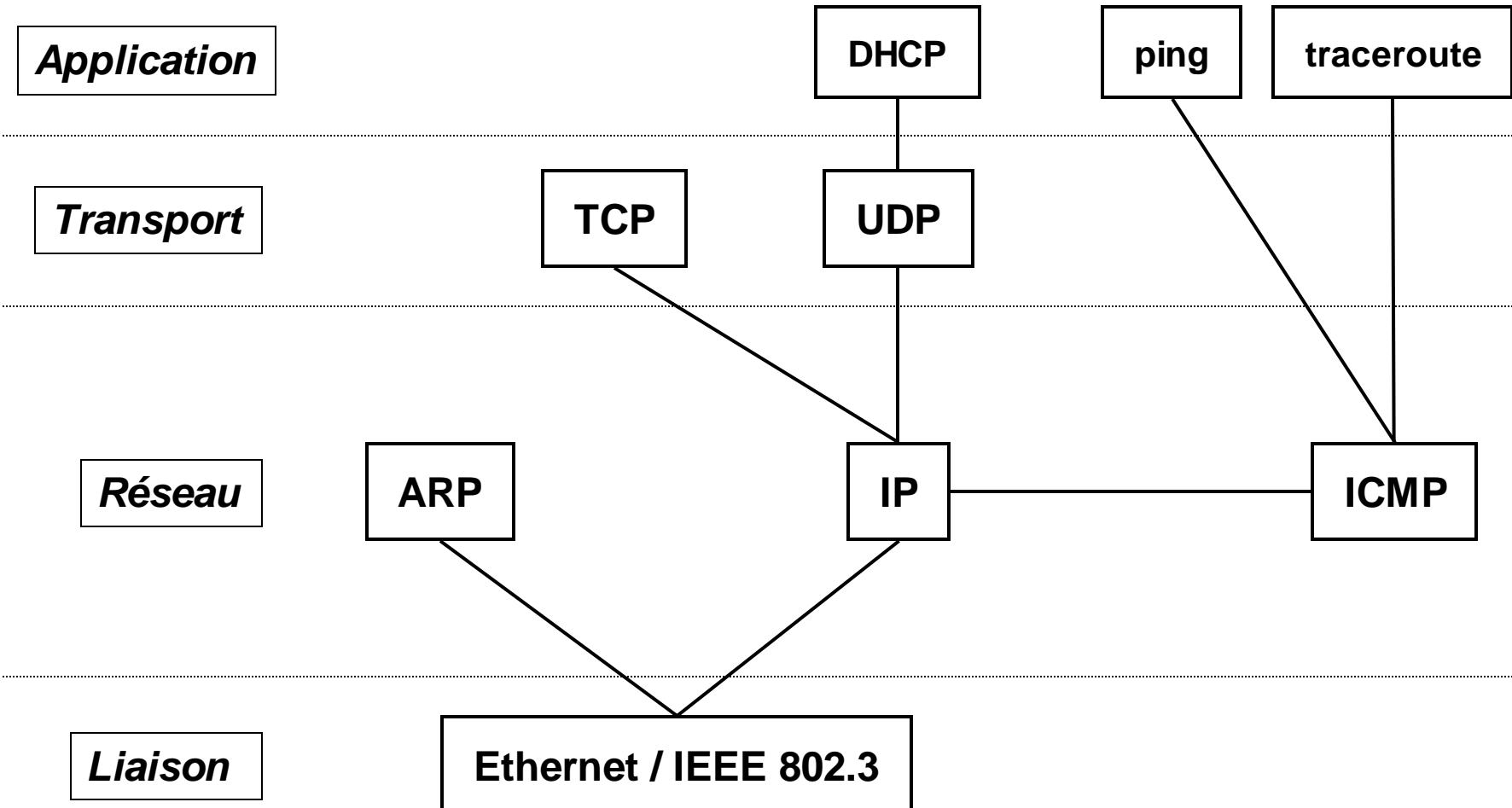
Couche réseau

- Assure l'acheminement des données de la source à la destination à travers un ou plusieurs des réseaux de communication intermédiaires entre les 2 systèmes terminaux.
- Unité d'information : le paquet
- Localisation des systèmes : adressage logique des nœuds et routage (trouve une route entre la source et la destination)
- Contrôle le flux des données acheminées : optimisation du réseau
- Déetecte et corrige les erreurs non réglées par la couche 2

La couche Réseau

- Acheminer les données entre l'émetteur et le destinataire à travers de différents réseaux en mettant en place un système **d'adressage hiérarchique**.
- C'est la première couche de **bout en bout**
- Les problèmes à traiter :
 - **Routage** : pour toutes paires d'adresses : trouver un chemin entre les 2 machines. Extension à un groupe d'adresses (diffusion, multicast). Routage.
 - **Annuaires** : Nommer (désigner) les machines : adresses réseaux, noms.

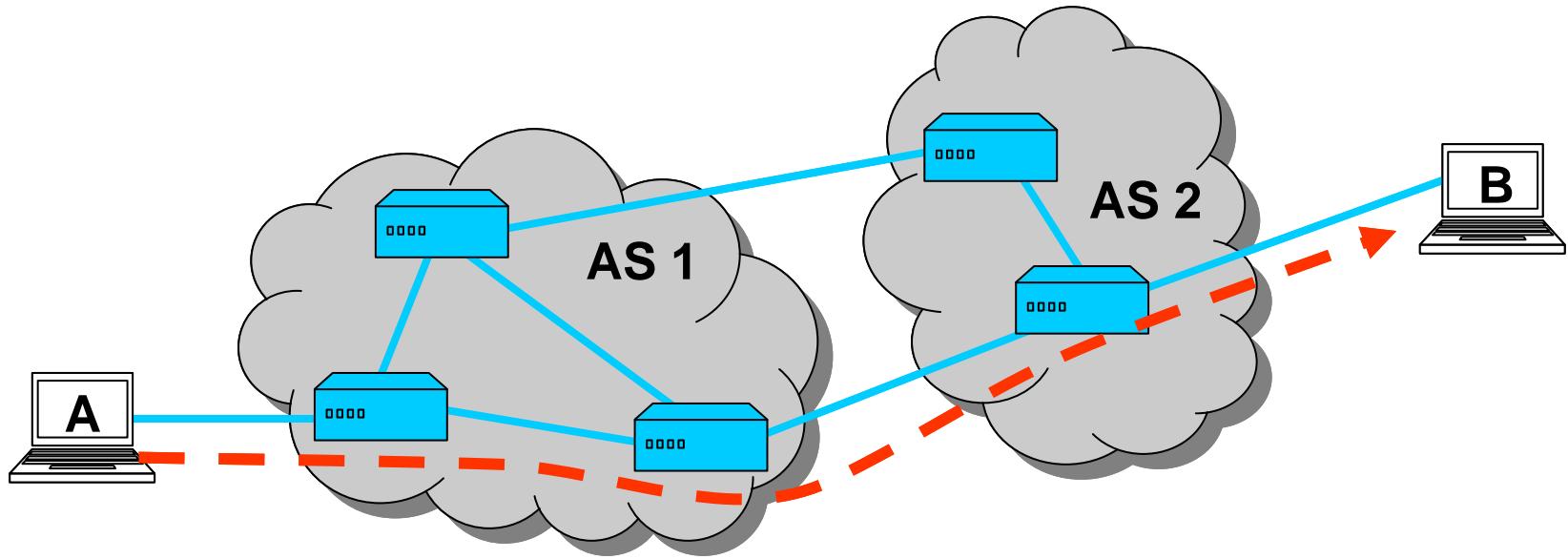
Couche 3 : architecture simplifiée



Qu'offre IP aux couches supérieures ?

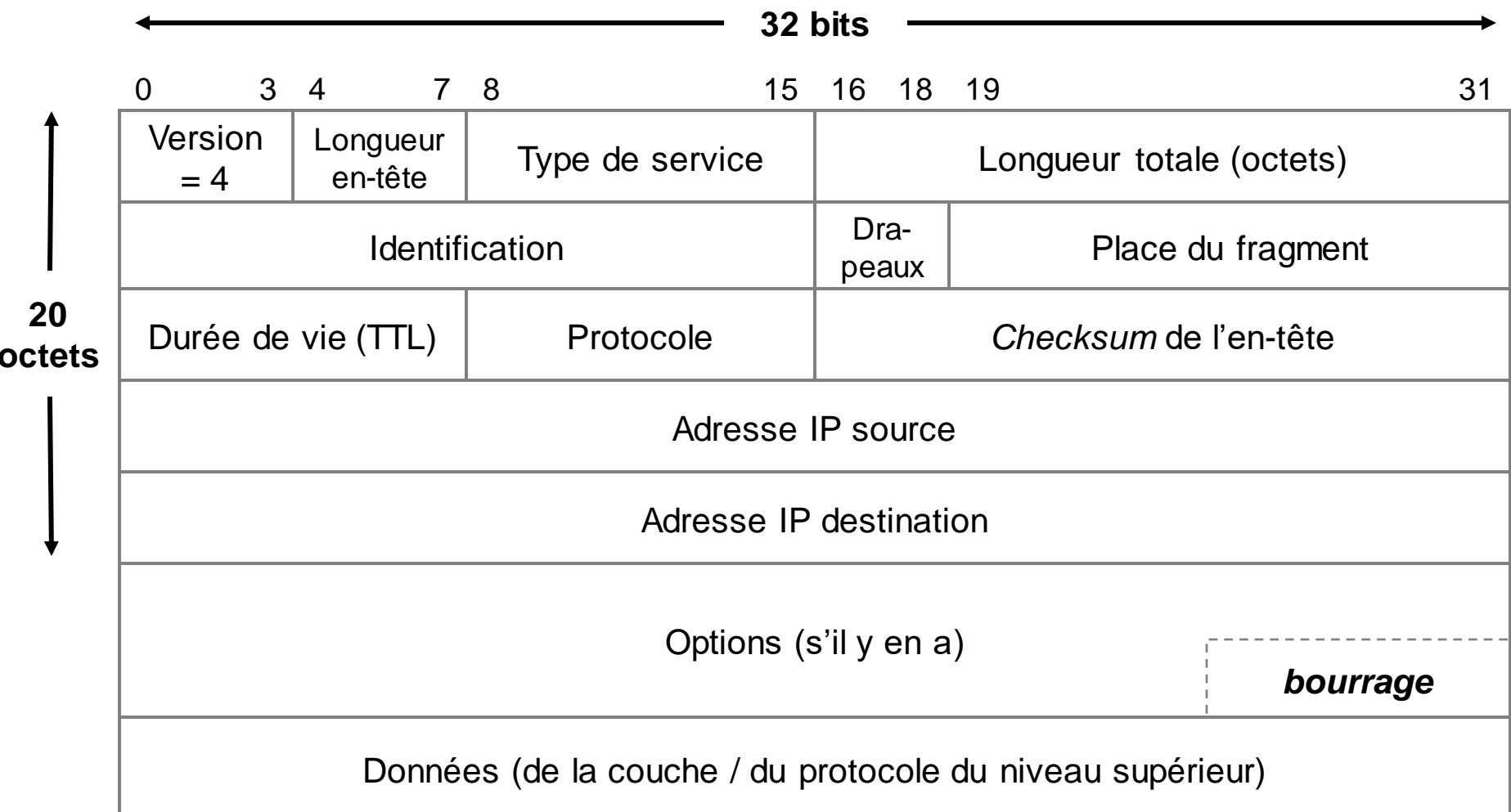
- Un service à « datagrammes »
 - Acheminement des paquets de la source à la (aux) destination(s)
 - Service non orienté connexion
 - Deux paquets consécutifs peuvent suivre une route différente pour aller de A à B
 - Service non fiable : « best effort »
 - Perte de paquets : possible
 - Duplication de paquets : possible
 - Arrivée des paquets en séquence : non garantie

Interconnexion de réseaux IP



- Adresses avec signification globale
- Relayage des paquets IP
 - Analyse des adresses (contenues dans l'en-tête IP)
 - Table de routage : quel est le prochain routeur sur le chemin ?

Format d'un paquet IPv4



Fragmentation

- La fragmentation d'un datagramme se fait au niveau des routeurs
- Si la MTU (*Maximum Transmission Unit*) de la liaison ne permet de transporter le paquet entier : envoi du paquet en fragments
- Le réassemblage est fait uniquement par le destinataire final
- Mécanisme coûteux pour les routeurs
- N'existe plus en IPv6

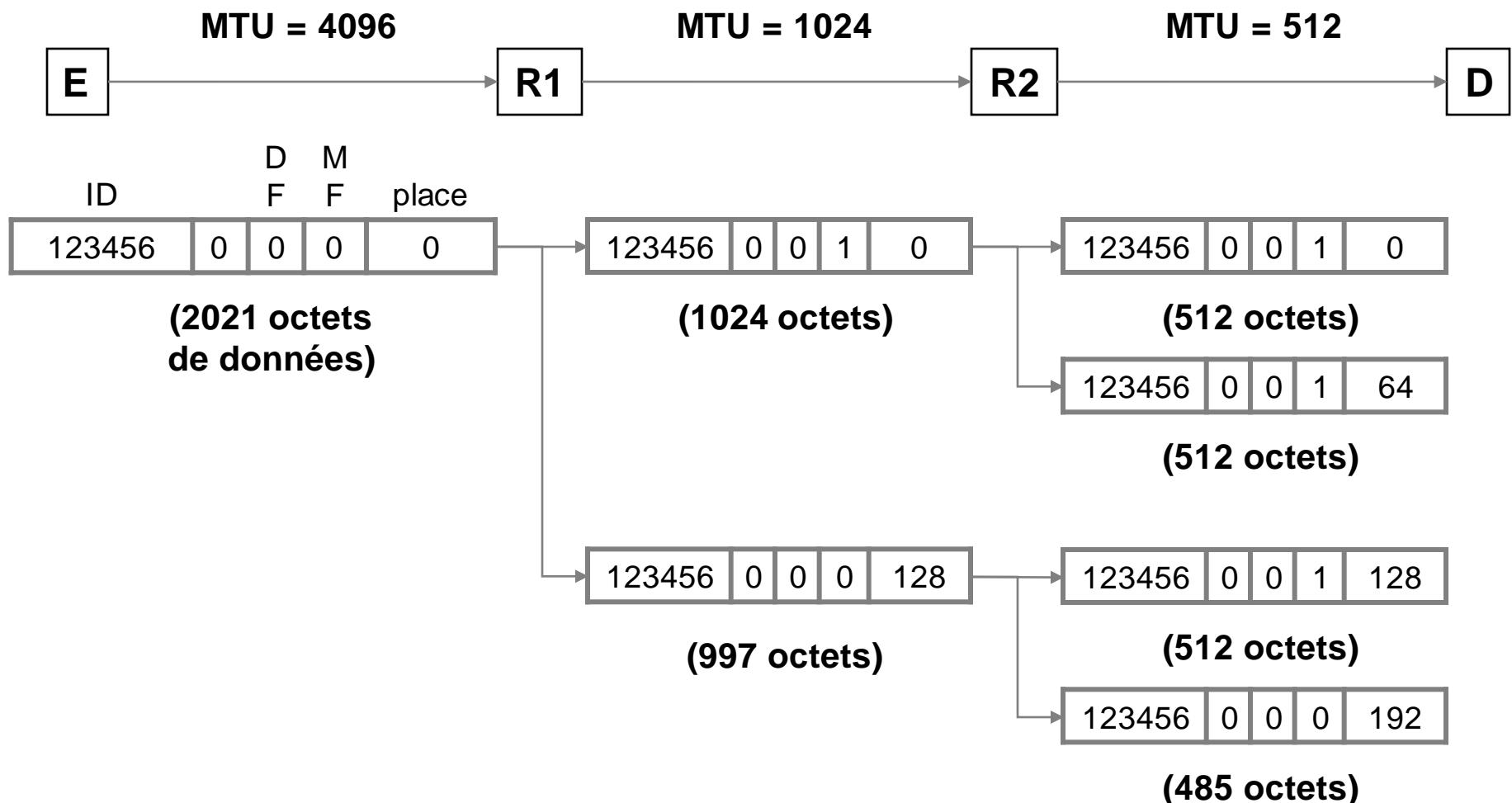


Fragmentation : champs de l'en-tête

- **Identification** : numéro unique (pour l'émetteur)
 - Si le paquet est fragmenté après, tous les fragments le portent
- **Place du fragment** : position du 1^{er} octet du fragment dans le datagramme original (non fragmenté)
 - Découpe des fragments en multiples de 8 octets
- **DF (don't fragment)** = 1 si le paquet ne doit pas être fragmenté
 - Si fragmentation nécessaire : écartement du paquet + génération d'un message ICMP vers la source
- **MF (more fragments)**
 - MF = 0 dernier fragment
- Drapeaux par défaut (paquet non fragmenté) : DF = MF = 0

16 bits	1 bit	1 bit	1 bit	13 bits
Identification	0	DF	MF	Place du fragment

Fragmentation : exemple



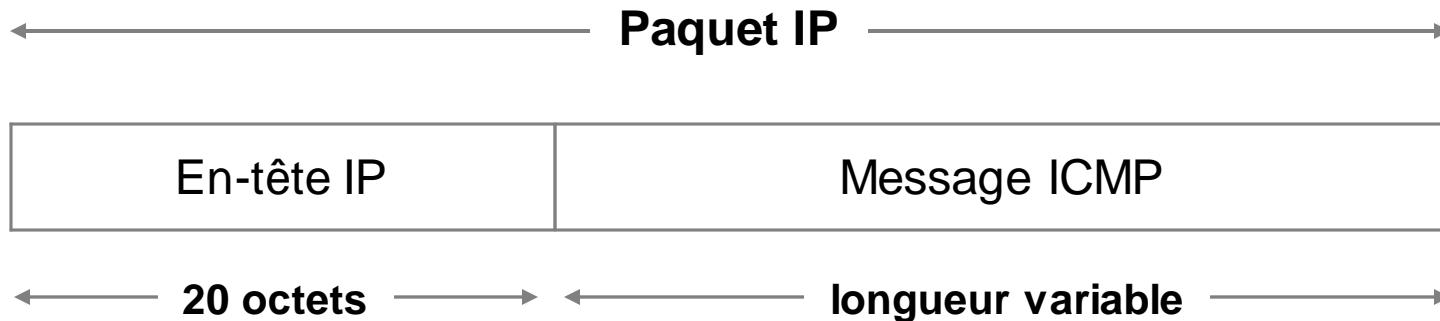
Champ Durée de vie (TTL)

- Initialisé à une valeur > 0
 - Valeur typique = 64
- Décrémenté d'une unité :
 - À chaque fois que le paquet traverse un routeur
 - Une fois/sec, si le paquet est en attente de réassemblage dans la station destinataire
- Quand TTL = 0, le paquet est détruit par le routeur et une notification (ICMP) est envoyé à l'émetteur du paquet
- **But :** éviter que les paquets bouclent indéfiniment dans le réseau si il y a des erreurs dans les tables de routage par exemple.

Couche réseau : protocole ICMP

Protocole ICMP

- *Internet Control Message Protocol* (RFC 792)
- But : échange de messages d'erreur et de demande d'information
 - Traités soit par IP, soit par une couche supérieure
- Niveau 3, mais encapsulé dans des paquets IP
 - Champ *Protocole* = 1



Commande *ping*

- Basée sur les messages ICMP de type 8 (*echo request*) et 0 (*echo reply*)
 - Réception d'un message type 8 => émission d'un message type 0
- Format des messages

0	7	8	15	16	23	24	31
Type (0 ou 8)		0		<i>Checksum</i>			
	Identificateur			Numéro de séquence			
(données optionnelles)							

**La réponse contient une copie des champs *Identificateur*,
N° de séquence et les données optionnelles**

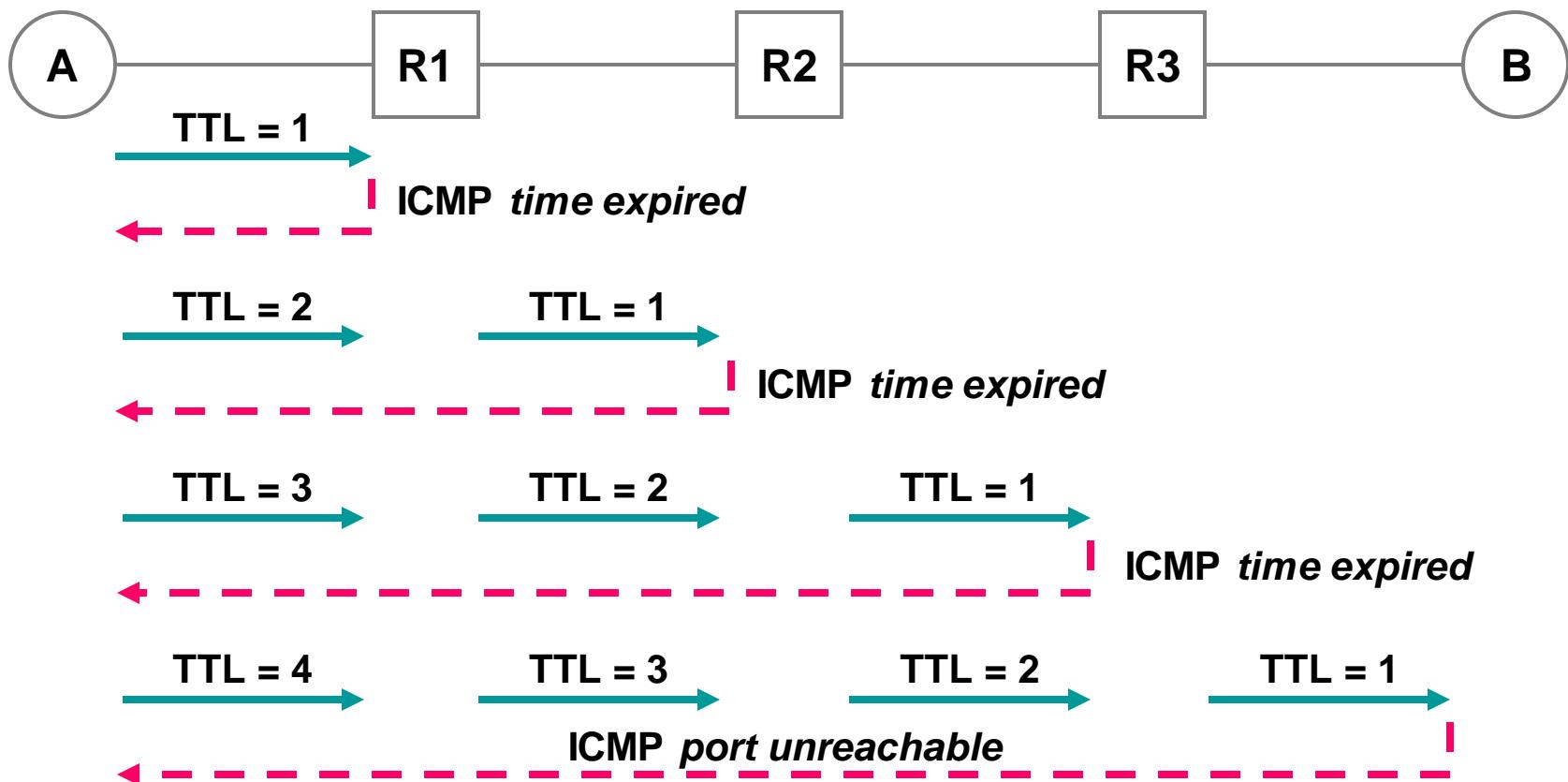
tracert

- Basé sur les messages ICMP de type 11 / code 0 (*time exceeded*) et type 3 / code 4 (*port unreachable*)
- En couche transport, il peut y avoir UDP, TCP ou ICMP

0	7 8	15 16	23 24	31
Type	Code	Checksum		
		0		
En-tête IP (y compris les options) + premiers 8 octets du paquet IP original				

tracer : fonctionnement

Sens A => B



tracert

```
Z:\>tracert www.google.fr
```

Détermination de l'itinéraire vers www.l.google.com [209.85.135.103]
avec un maximum de 30 sauts :

1	1 ms	<1 ms	<1 ms	sophia-p32.inria.fr [138.96.32.250]
2	<1 ms	<1 ms	<1 ms	nice-g3-0-60.cssi.renater.fr [193.51.181.138]
3	15 ms	15 ms	15 ms	marseille-pos4-0.cssi.renater.fr [193.51.179.245]
4	15 ms	15 ms	15 ms	montpellier-pos2-0.cssi.renater.fr [193.51.179.242]
5	15 ms	15 ms	15 ms	lyon-pos14-0.cssi.renater.fr [193.51.179.221]
6	14 ms	14 ms	14 ms	nri-b-pos9-0.cssi.renater.fr [193.51.179.13]
7	15 ms	15 ms	15 ms	TELEGLOBE-FRANCE-INTERNATIONAL.sphinx.tm.fr [194.68.129.242]
8	15 ms	15 ms	15 ms	if-6-0-7.core1.PU1-Paris.teleglobe.net [80.231.79.14]
9	24 ms	24 ms	24 ms	if-2-0-0.core2.FR1-Frankfurt.teleglobe.net [80.231.65.65]
10	32 ms	35 ms	35 ms	12.icore1.FR1-Frankfurt.teleglobe.net [80.231.65.6]
11	24 ms	25 ms	25 ms	195.219.180.30
12	25 ms	49 ms	25 ms	209.85.249.178
13	31 ms	31 ms	31 ms	209.85.248.248
14	31 ms	36 ms	31 ms	72.14.239.46
15	38 ms	35 ms	33 ms	72.14.239.58
16	32 ms	31 ms	32 ms	mu-in-f103.google.com [209.85.135.103]

Itinéraire déterminé.

```
Z:\>_
```

Path MTU

- Découverte de la taille maximale des paquets au long de la route A => B pour éviter la fragmentation
 - Émission avec le bit DF = 1 (en-tête IP)
 - Si un routeur doit fragmenter, il retourne à la source un message d'erreur ICMP :

0	7	8	15	16	23	24	31
Type = 3		Code = 4			Checksum		
	0				MTU requis		
En-tête IP (y compris les options) + premiers 8 octets du paquet IP original							

Couche Réseau : Adressage

Adressage IPv4

- 32 bits (4 octets). Notation classique : 4 octets en décimal séparés par des '.'
- **Exemple** : 194.214.171.200
- La composition des 32 bits :
 - identification du réseau (net-id)
 - identification de la machine (host-id)
- **Taille net-id/host-id** : selon la classe de l'adresse (et le masque).
- Pour garantir l'unicité et le routage, l'**ICANN** (*Internet Corporation for Assigned Names and Numbers*), est chargée d'attribuer des adresses IP publiques (les adresses IP des ordinateurs directement connectés sur le réseau public internet).
- La partie host-id est assignée par le gestionnaire du réseau local

Adresses IP spéciales

- **<net-id><0>** : on obtient **l'adresse réseau**. Cette adresse ne peut être attribuée à aucun des ordinateurs du réseau.
- **<0><host-id>** : on obtient **l'adresse machine**. Cette adresse représente la machine spécifiée par le host-ID qui se trouve sur le réseau courant.
- **<net-id><1>** : on obtient **l'adresse de diffusion** (en anglais **broadcast**). Pour envoyer à toutes les machines situées sur le réseau spécifié par le net- id.
- **<1><0>** : le masque où le nombre de 1 définit la taille du net-id
- **0.0.0.0** : dans les routeurs, route par défaut.
- Enfin, l'adresse **127.0.0.1** désigne la **machine locale** (en anglais **localhost**).

Exemple

- 192.168.1.102 avec masque 255.255.255.0
- Taille du net-id = *nb de 1 du masque* = ??
- Taille du host-id = $32 - |\text{net-id}|$ = ??
- Broadcast = $\langle \text{net-id} \rangle \langle 1 \rangle$ = ??
- Adresse réseau = $\langle \text{net-id} \rangle \langle 0 \rangle$ = ??
- Nombre de machines sur ce réseau = $2^{|\text{host-id}|} - 2$ =
- Plage d'adresses =

Aide : $255_{(10)} = 1111\ 1111_{(2)}$ soient 8 « 1 »

Exemple

- 192.168.1.102 avec masque 255.255.255.0
- Taille du net-id = *nb de 1 du masque* = 24
- Taille du host-id = $32 - |\text{net-id}|$ = 8
- Broadcast = $\langle \text{net-id} \rangle \langle 1 \rangle$ = **192.168.1.255**
- Adresse réseau = $\langle \text{net-id} \rangle \langle 0 \rangle$ = **192.168.1.0/24**
- Nombre de machines sur ce réseau = $2^{|\text{host-id}|} - 2$
= 254
- De 192.168.1.1 à 192.168.1.254

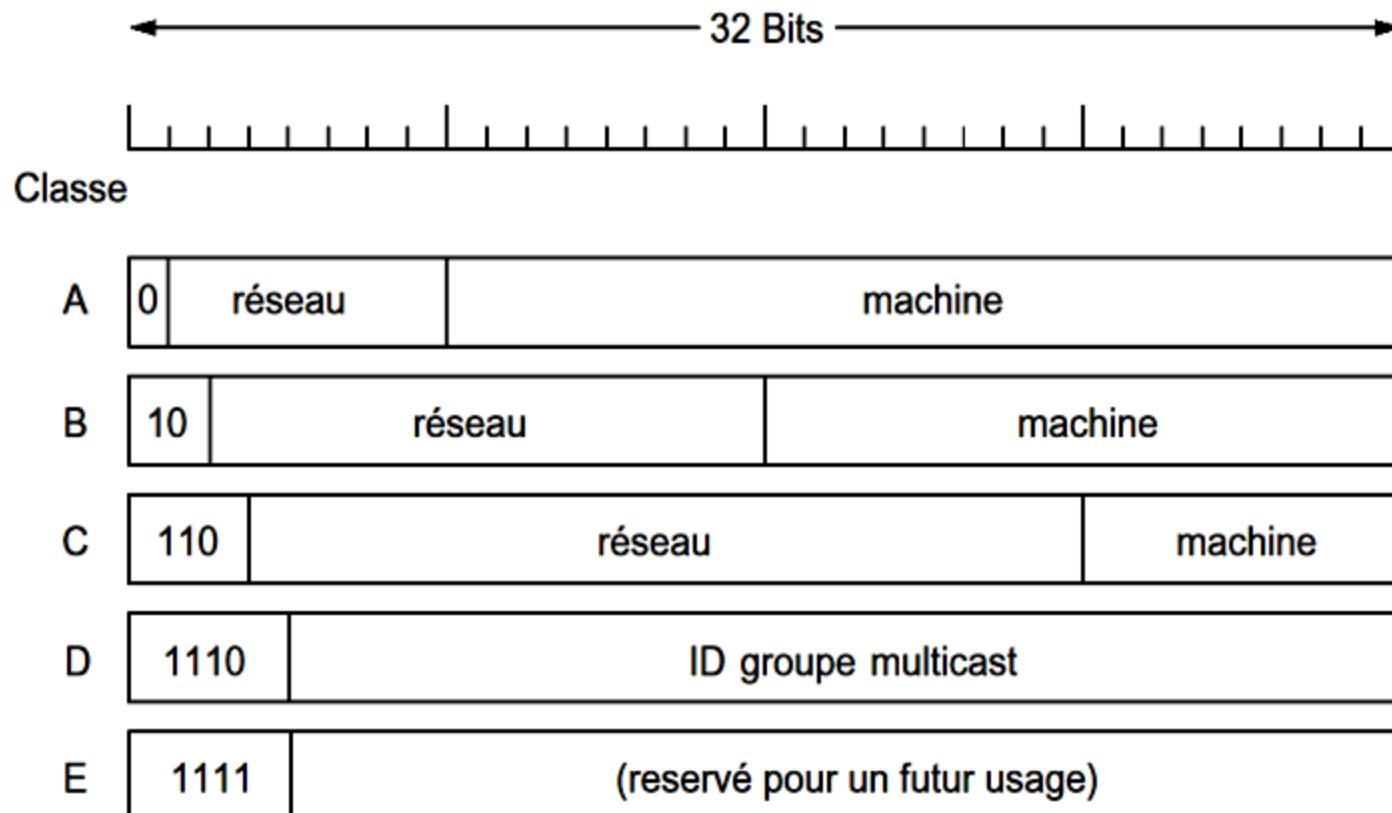
134.59.1.3/16, |net-id|=16

- **Masque** =
 - **@réseau** =
 - **@broadcast** =
 - **Nombre de machines** =
 - **Plage d'adresses** = de à

134.59.1.3/16, $|net-id|=16$

- **Masque** = 11111111.11111111.00000000.00000000
= 255.255.0.0
- **@réseau** = 134.59.0.0
- **@broadcast** = 134.59.255.255
- **Nombre de machines** = $2^{16}-2 = 65534$ ($2^{|host-id|} - 2$)
- **Plage d'adresses** = de 134.59.0.1 à 134.59.255.254

Classes d'adresses IP (avant 1994)



Classe A

de	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1.0.0.1
à	0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0	126.255.255.254

De 1.0.0.1 à 126.255.255.254

- net-id = 1^{er} octet
- 1^{er} bit de poids fort à 0
- Adressage de 126 réseaux (2^7 -2), chacun pouvant contenir plus de 16 millions (2^{24} -2), de machines.
- Masque = 255.0.0.0

Classe B

de	1 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 1	128.0.0.1
à	1 0 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 0	191.255.255.254

De 128.0.0.1 à 191.255.255.254

- net-id = deux premiers octets
- Les 2 bits de poids forts = 10
- Adressage de 16384 réseaux (2^{14}) chacun pouvant contenir 65534 ($2^{16}-2$) machines
- C'est la classe la plus utilisée, les adresses aujourd'hui sont pratiquement épuisées.
- Masque = 255.255.0.0

Classe C

de **1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1** 192.0.0.1
à **1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0** 223.255.255.254

De 192.0.0.1 à 223.255.255.254

- net-id = les trois premiers octets, les 3 bits de poids forts sont à 110
- Adressage de plus de 2 millions de réseaux (2^{21}), chacun pouvant contenir 254 machines.
- Masque = 255.255.255.0

Classe D

1 1 1 0

adresse multicast

De 224.0.0.1 à 239.255.255.254

- Les 4 bits de poids forts sont à 1110
- Adresse de diffusion vers les machines d'un même groupe qui se sont abonnées

Adressage IP (avant 1994)

- Espace d'adressage plat
 - Pas de numérotation hiérarchique
 - Pas de rapport entre adresse et localisation géographique : privilégier la simplicité d'administration
- Classes A, B, C : utilisation inefficace et peu flexible des adresses
- Évolutions :
 - CIDR
 - Adresses privées + NAT
 - IPv6 (adresses sur 128 bits)

CIDR (Classless Inter-Domain Routing)

- Mis au point afin (principalement) de diminuer la taille de la table de routage contenue dans les routeurs
- Agréger plusieurs entrées de cette table en une seule (par région géographique et fournisseurs d'accès)
- Agrégation maximum des sous-réseaux qui sont routés ensemble avec la même politique
- L'adresse IP est suivie par un slash ("/") indiquant le nombre de bits correspondant au net-id
- **Exemple :** 192.0.2.96/23 indique une adresse IP où les 23 premiers bits sont utilisés comme adresse réseau.

Le masque comporte 23 '1' suivis de 9 '0': 255.255.254.0

Rappel : 1111 1111 = 255

Exemple : 192.44.77.0 /26

- **MASQUE** : 26 ‘un’ suivis de 6 ‘zéro’ ($8 \text{ ‘un’} = 255$)
- **ADRESSE RESEAU** : je recopie les 26 bits de l’adresse, je complète par des ‘zéro’
- **BROADCAST** : je recopie les 26 bits de l’adresse, je complète par des ‘un’
- **NOMBRE DE MACHINES** : $2^{\lceil \text{host-id} \rceil} - 2$
- **PLAGE D’ADRESSES** : De ‘@réseau +1’ à ‘@Broadcast -1’

Exemple : 192.44.77.0 /26

- **MASQUE** : 26 ‘un’ suivis de 6 ‘zéro’ (8 ‘un’ = 255)
 - 255.255.255.11000000 = 255.255.255.192
- **ADRESSE RESEAU** : je recopie les 26 bits de l’adresse, je complète par des ‘zéro’
 - 192.44.77.00000000 = 192.44.77.0
- **BROADCAST** : je recopie les 26 bits de l’adresse, je complète par des ‘un’
 - 192.44.77.00111111 = 192.44.77.63
- **NOMBRE DE MACHINES** : $2^{\lceil \text{host-id} \rceil} - 2$
 - $2^6 - 2 = 62$
- **PLAGE D'ADRESSES** : De '@réseau +1' à '@Broadcast -1'
 - De 192.44.77.1 à 192.44.77.62

Appartenance d'une adresse à un réseau => on cherche l'@ réseau

Exemple : réseau 130.79.152.0/23

Est-ce que l'adresse 130.79.154.1 fait partie de ce réseau ?

130.79.154.1/23 → @réseau =

Et l'adresse 130.79.153.35 ?

130.79.153.35/23 → @réseau =

152 = **10011000**

153 = **10011001**

154 = **10011010**

Appartenance d'une adresse à un réseau => on cherche l'@ réseau

Exemple : réseau 130.79.152.0/23

Est-ce que l'adresse 130.79.154.1 fait partie de ce réseau ?

130.79.154.1/23 → @réseau = 130.79.154.0

Et l'adresse 130.79.153.35 ?

130.79.153.35/23 → @réseau = 130.79.152.0

152 = 10011000

153 = 10011001

154 = 10011010

Subnetting

- Diviser un **gros réseau** unitaire en ce qui apparaît comme **plusieurs sous-réseaux**
- Les sous-réseaux sont utiles pour réduire le nombre d'entrées dans la table de routage pour Internet en cachant des informations sur les sous-réseaux individuels d'un site
- De plus, cela a permis de réduire la surcharge réseau (overhead), en divisant le nombre d'hôtes recevant des appels broadcast IP

Subnetting

- **Masque de sous-réseau :** indique le nombre de bits utilisés pour identifier le sous-réseau, et le nombre de bits caractérisant les hôtes.
- Un masque de sous réseau est une adresse de 32 bits contenant des 1 aux emplacements des bits que l'on désire conserver, et des 0 pour ceux que l'on veut rendre égaux à zéro.
- Une fois ce masque créé, il suffit de faire un ET entre la valeur que l'on désire masquer et le masque afin de garder intacte la partie que l'on désire et annuler le reste
- Il y a plusieurs avantages à utiliser ce procédé. Un d'entre eux est de pouvoir connaître le réseau associé à une adresse IP.

Exemple de création de sous-réseaux

- 130.79.153.28/23 : adresse réseau 130.79.152.0
- Adresse réseau : 10000010 01001111 10011000 00000000
- Si je veux créer 3 sous-réseaux : j'ai besoin de 2 bits supplémentaires dans l'adresse réseau pour les créer (je crée ainsi 4 sous-réseaux).
 1. 10000010 01001111 10011000 00000000 : 130.79.152.0/25
 2. 10000010 01001111 10011000 10000000 : 130.79.152.128/25
 3. 10000010 01001111 10011001 00000000 : 130.79.153.0/25
 4. 10000010 01001111 10011001 10000000 : 130.79.153.128/25
- Chaque sous-réseau pourra adresser 126 hôtes.