# Couche Réseau
# IP & ICMP

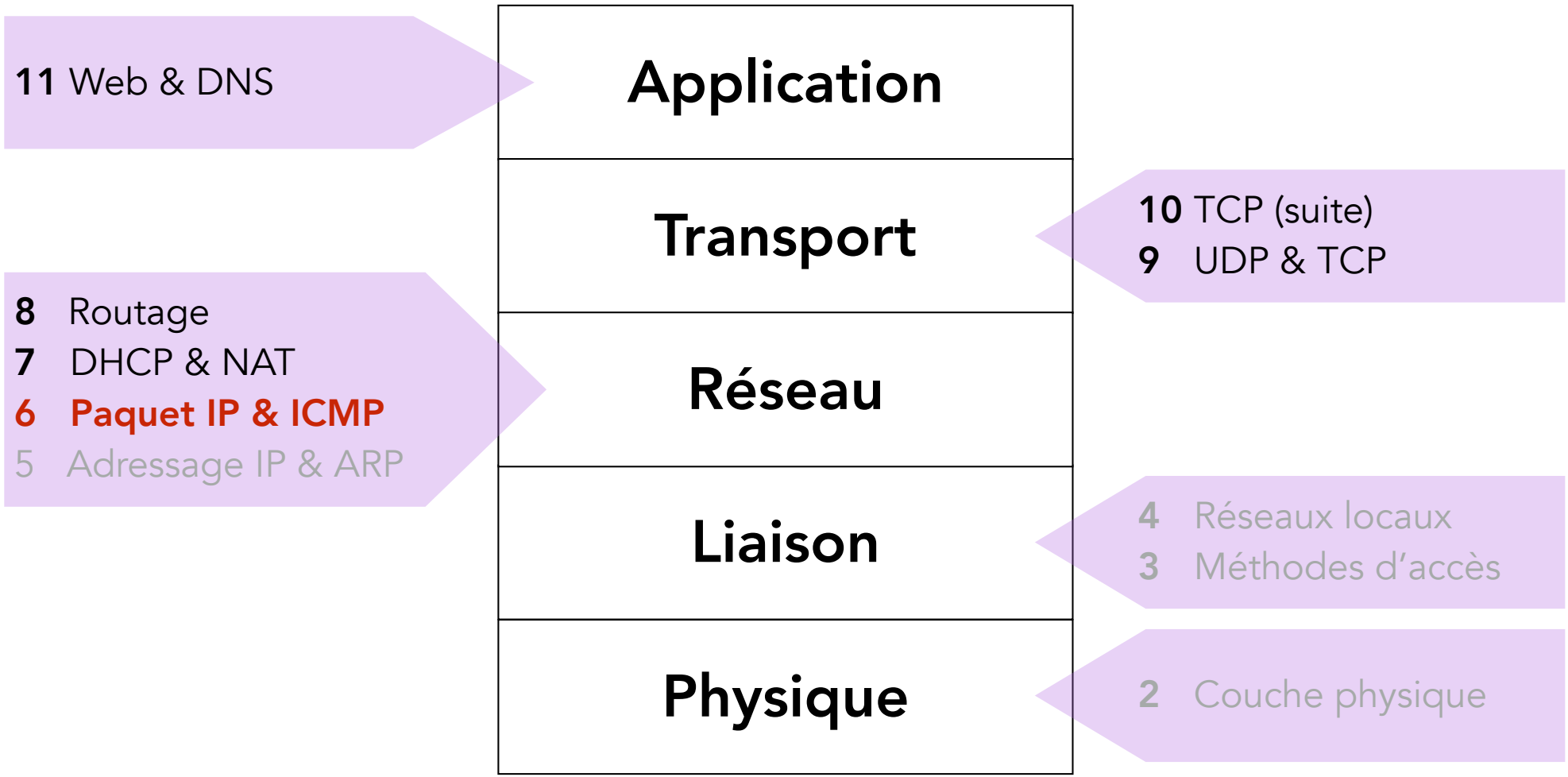
UE LU3IN033 Réseaux

2024-2025

**Bruno Baynat**

Bruno.Baynat@sorbonne-universite.fr

# Programme de l'UE LU3IN033

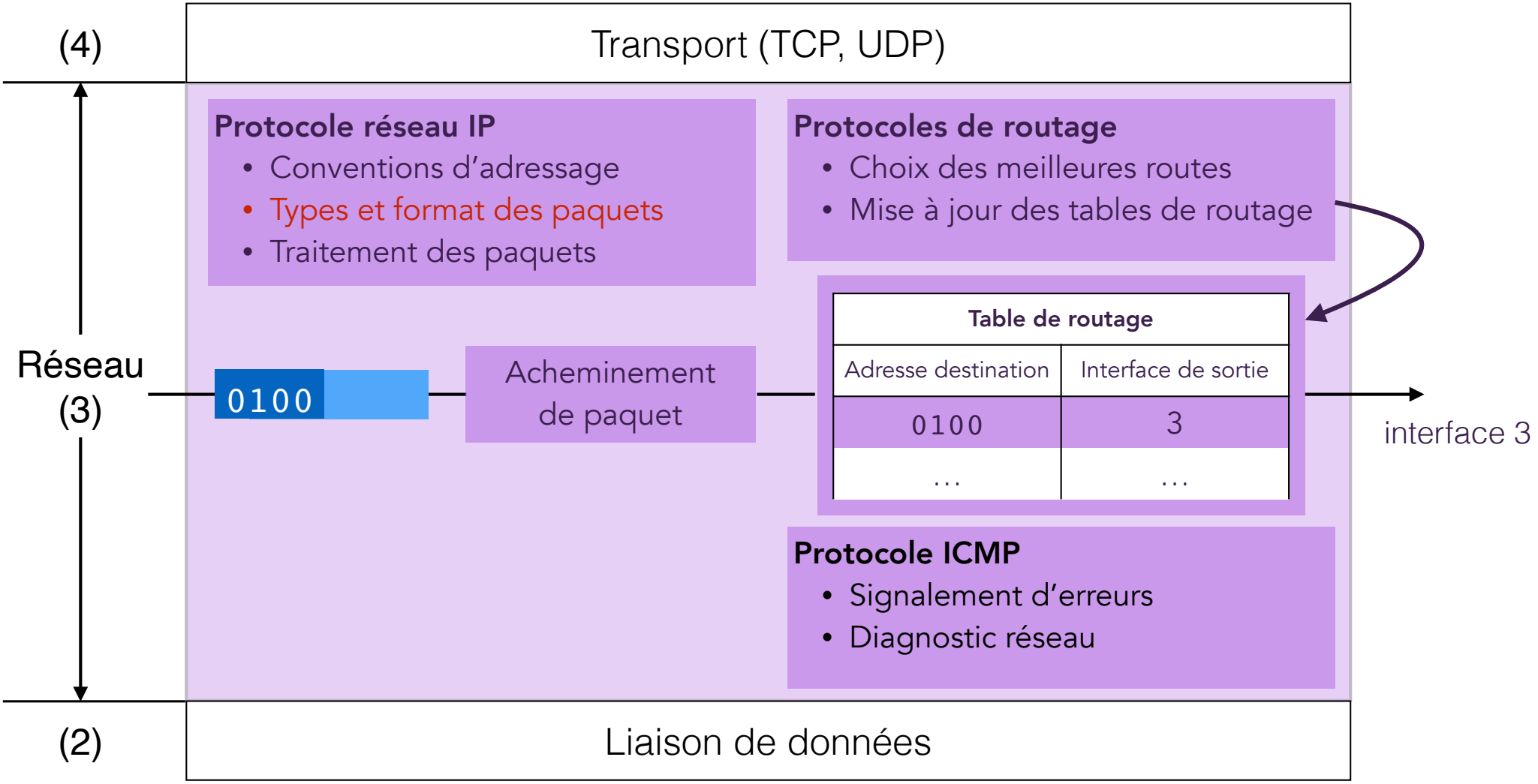


# Plan du cours

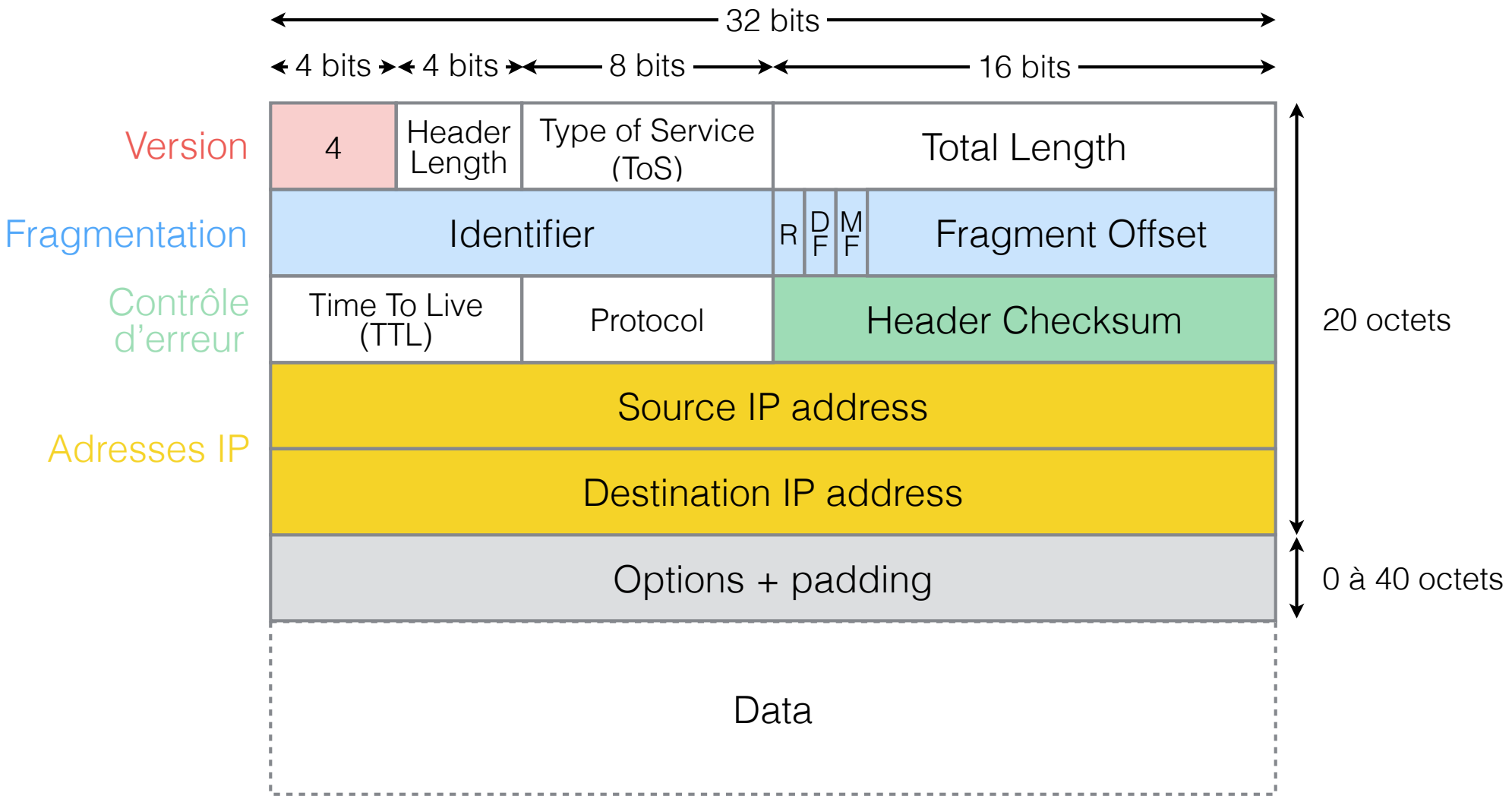
- Entête du paquet IP
  - Champs de la partie fixe
  - Options IP
  - Contrôle d'erreur
- Longueur d'un paquet IP
  - Taille maximale
  - Fragmentation
- Charge utile du paquet IP
  - Protocoles encapsulés
- Acheminement IP
  - Acheminement direct vs indirect
  - Table d'acheminement
- Protocole ICMP
  - Tests et diagnostic d'erreurs
  - Commandes ping et traceroute

# Paquet IP

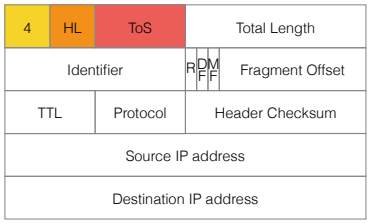
# Couche réseau



# Paquet IPv4

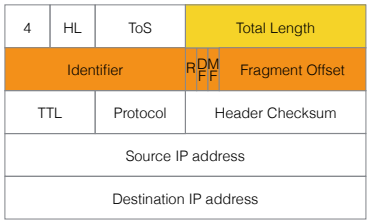

32 bits
4 bits
4 bits
8 bits
16 bits
Version
Fragmentation
Contrôle d'erreur
Adresses IP
4
Header Length
Type of Service (ToS)
Total Length
Identifier
R
DF
MF
Fragment Offset
Time To Live (TTL)
Protocol
Header Checksum
Source IP address
Destination IP address
Options + padding
Data
20 octets
0 à 40 octets

# Champs de l'entête IP




- Version (4 bits)
  - Indique la version du protocole IP
  - Nécessaire pour déterminer la structure de l'entête du paquet
    - valeurs courantes : "4" (pour IPv4) et "6" (pour IPv6)
- Header Length (4 bits)
  - Taille de l'entête exprimée en nombre de mots de 32 bits (4 octets)
    - valeur min "5" (0101) : 20 octets (pas d'option IP)
    - valeur max "15" (1111) : 60 octets (40 octets d'options IP)
- Type-of-Service (8 bits)
  - Conçu historiquement pour permettre aux routeurs de différencier les types de trafic et de leur attribuer des priorités
  - N'est plus utilisé de nos jours (ToS mis à zéro)
    - ou remplacé par le champ DSCP (*Differentiated Services Code Point*)

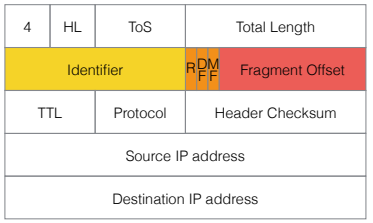
# Champs de l'entête IP



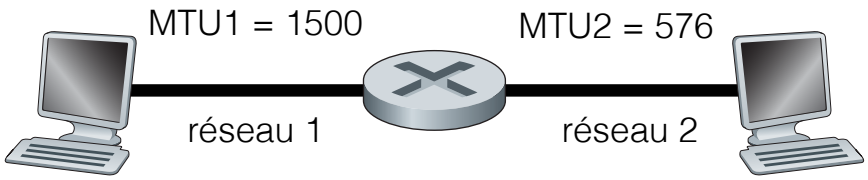

- Total Length (16 bits)
  - Taille totale du paquet exprimée en octets
    - taille max d'un paquet : 65535 octets ($2^{16}$ -1)
  - En pratique la taille d'un paquet est limitée par la MTU
    - *Maximum Transmission Unit*
    - taille maximale du champ données des trames utilisées par la couche liaison de données sous-jacente
    - Ex : MTU des trames Ethernet = 1500 octets
- Fragmentation (32 bits)
  - Permet de gérer la fragmentation d'un paquet et le réassemblage des fragments
  - Tous les fragments issus d'un même paquet possèdent le même identifiant

# Fragmentation IP

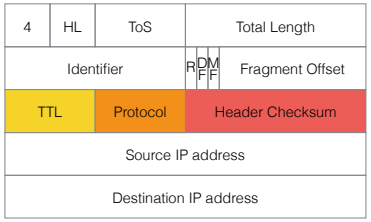
MTU (*Maximum Transmission Unit*) : taille maximale du champ données des trames utilisées par la couche liaison de données sous-jacente




- Flags : (3 bits : 0, DF, MF)
  - DF : Don't Fragment
  - MF : More Fragment
- Fragment Offset (13 bits)
  - Taille (en octets) des données des fragments précédant le fragment courant divisée par 8
- Exemple
  - Données IP : 1300 octets
  - Pas de fragmentation sur le réseau 1
  - Fragmentation sur le réseau 2 :
    - une trame peut transporter au maximum 576 octets de données = 20 octets d'en-tête IP + 556 octets de données IP
    - valeur multiple de 8 la plus proche : 552 = 69 * 8
    - fragment 1 : DF = 0 MF = 1 offset = 0 (données : 552 octets)
    - fragment 2 : DF = 0 MF = 1 offset = 69 = 552/8 (données : 552 octets)
    - fragment 3 : DF = 0 MF = 0 offset = 138 = 1104/8 (données : 196 octets)

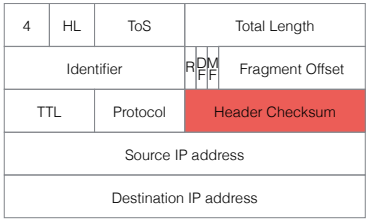


# Champs de l'entête IP




- Time-To-Live (8 bits)
  - Nombre maximal de sauts autorisé sur le chemin emprunté
  - Valeur décrémentée de 1 par chacun des routeurs que traverse le paquet
    - suppression du paquet dont le TTL est à 0
    - envoi à la source d'un message ICMP *Time exceeded*
- Protocol (8 bits)
  - Identifie le protocole du message encapsulé dans le paquet IP
    - 1 : ICMP
    - 6 : TCP
    - 17 : UDP
- Header Checksum (16 bits)
  - Code de détection d'erreurs portant sur l'entête
  - Vérification de bout en bout
    - la source calcule la valeur du checksum du paquet envoyé
    - le récepteur vérifie la valeur du checksum du paquet reçu

# Exemple : calcul du checksum



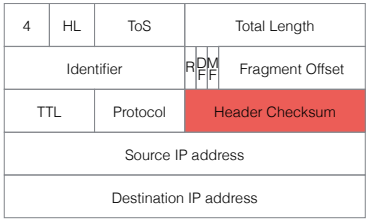

entête Ethernet | entête IP | entête TCP | données TCP

```
08 00 20 87 b0 44 08 00 11 08 c0 63 08 00 45 00
00 48 49 ba 00 00 1e 06 69 8d c1 37 33 f6 c1 37
33 04 17 70 96 d4 39 7f 84 c2 bf 3a 21 fd 50 18
11 1c 99 bc 00 00 0e 00 31 3f 02 c0 00 11 00 00
3e c1 00 00 00 11 00 00 00 02 28 28 a7 b0 80 29
ea fc 81 58 90 70
```

16 bits

```
0x4500      0100 0101 0000 0000
0x0048      0000 0000 0100 1000
            0100 0101 0100 1000
0x49ba      0100 1001 1011 1010
            1000 1111 0000 0010
0x0000      0000 0000 0000 0000
0x1e06      0001 1110 0000 0110
            1010 1101 0000 1000
0x0000      0000 0000 0000 0000
0xc137      1100 0001 0011 0111
          1 0110 1110 0011 1111
0x33f6      0011 0011 1111 0110
          1 1010 0010 0011 0101
0xc137      1100 0001 0011 0111
         10 0110 0011 0110 1100
0x3304      0011 0011 0000 0100
-----------------------------------
         10 1001 0110 0111 0000
            1001 0110 0111 0010
0x698d      0110 1001 1000 1101
```

# Exemple : vérification du checksum




| entête Ethernet | entête IP | entête TCP | données |
|---|---|---|---|

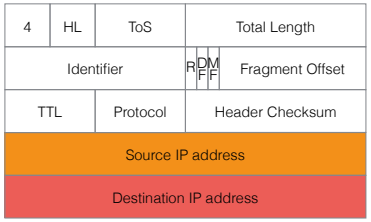
```
08 00 20 87 b0 44 08 00 11 08 c0 63 08 00 45 00
00 48 49 ba 00 00 1e 06 69 8d c1 37 33 f6 c1 37
33 04 17 70 96 d4 39 7f 84 c2 bf 3a 21 fd 50 18
11 1c 99 bc 00 00 0e 00 31 3f 02 c0 00 11 00 00
3e c1 00 00 00 11 00 00 00 02 28 28 a7 b0 80 29
ea fc 81 58 90 70
```

16 bits

```
0x4500    0100 0101 0000 0000
0x0048    0000 0000 0100 1000
          0100 0101 0100 1000
0x49ba    0100 1001 1011 1010
          1000 1111 0000 0010
0x0000    0000 0000 0000 0000
0x1e06    0001 1110 0000 0110
          1010 1101 0000 1000
0x698d    0110 1001 1000 1101
        1 0001 0110 1001 0101
0xc137    1100 0001 0011 0111
        1 1101 0111 1100 1100
0x33f6    0011 0011 1111 0110
       10 0000 1011 1100 0010
0xc137    1100 0001 0011 0111
       10 1100 1100 1111 1001
0x3304    0011 0011 0000 0100
       10 1111 1111 1111 1101
                          10
          1111 1111 1111 1111
```

16 bits à 1 : entête sans erreur

# Adresses source et destination




- Deux adresses IP
  - Adresse IP de la source (32 bits)
  - Adresse IP de la destination (32 bits)
- Adresse destination
  - Identifie la machine hôte destination
  - Utilisée par les routeurs pour acheminer le paquet
  - Résulte fréquemment de la résolution du nom d'un serveur (DNS)
- Adresse source
  - Identifie la machine hôte source
  - Permet à la destination de savoir d'où vient le paquet qu'elle reçoit
  - Utilisée par la destination pour répondre à la source
  - Configurée manuellement (administrateur) ou découverte dynamiquement (DHCP)

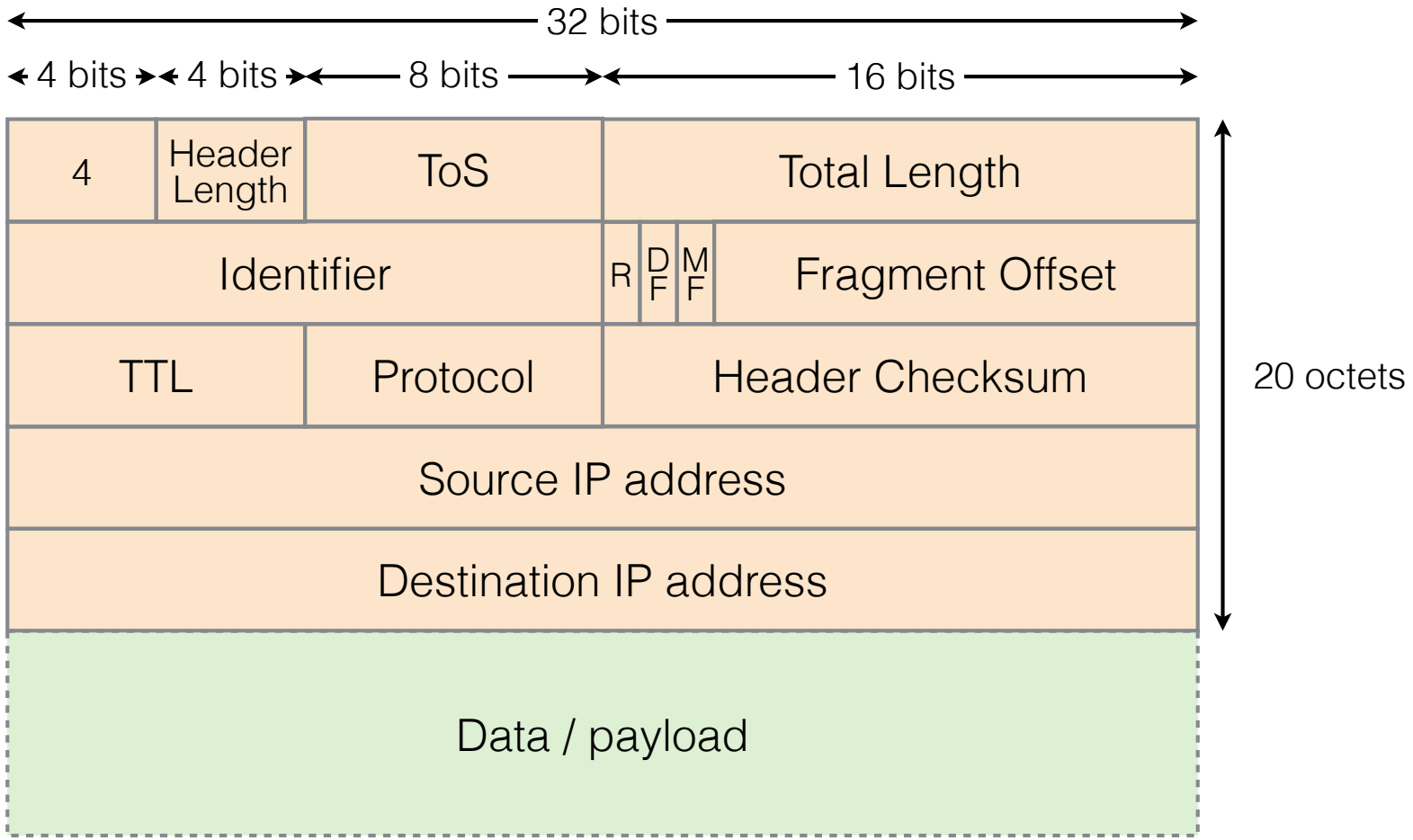
# Exemple de trace

| | IHL * 4 = | Total length - (IHL * 4) = |
|---|---|---|
| 14 octets | 20 octets | 52 octets |
| entête Ethernet | entête IP | données IP |

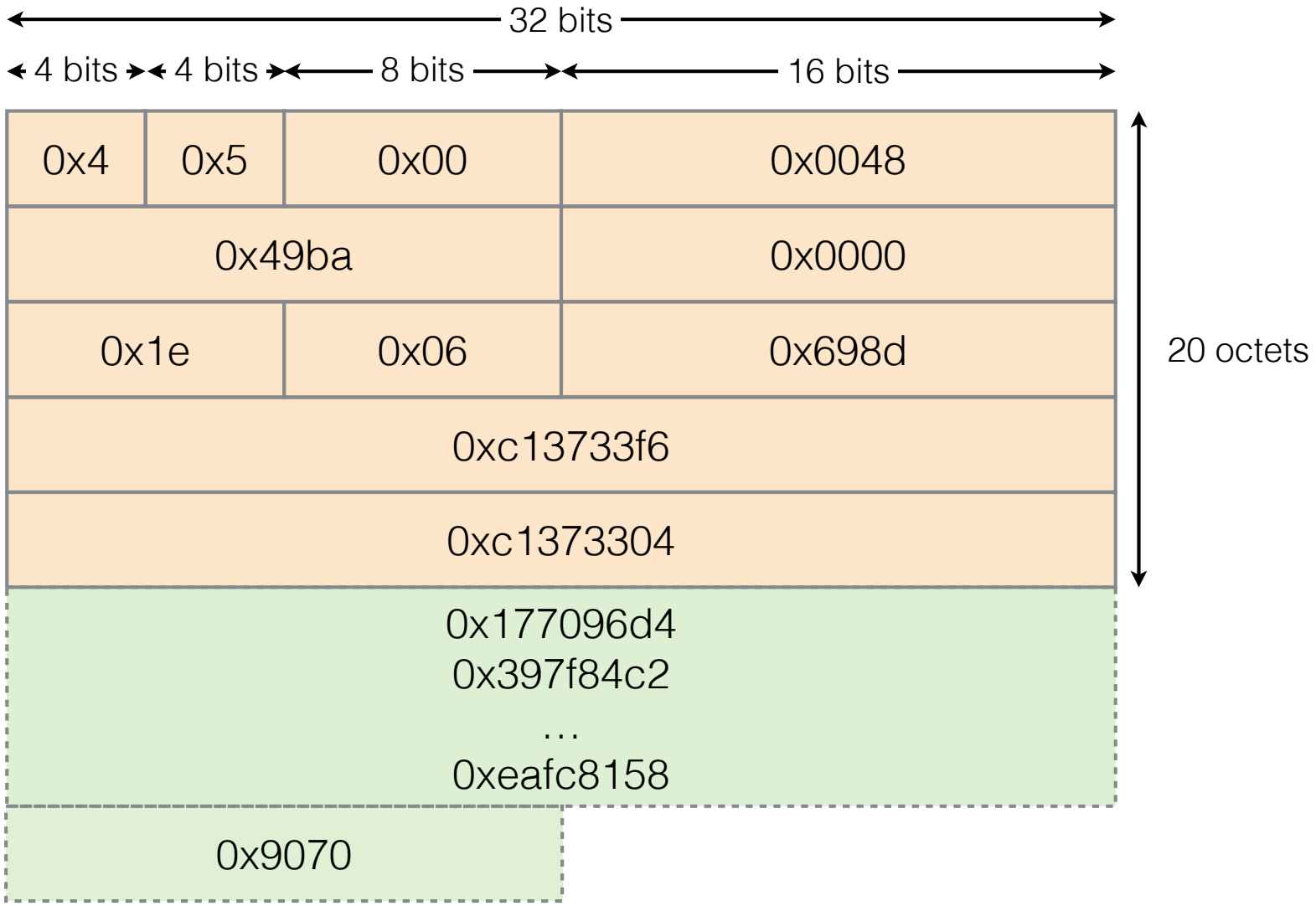
numéro en hexa de l'octet en début de ligne

```
0x00  08 00 20 87 b0 44 08 00 11 08 c0 63 08 00 45 00   octets 0 à 15
0x10  00 48 49 ba 00 00 1e 06 69 8d c1 37 33 f6 c1 37   octets 16 à 31
0x20  33 04 17 70 96 d4 39 7f 84 c2 bf 3a 21 fd 50 18   octets 32 à 47
0x30  11 1c 99 bc 00 00 0e 00 31 3f 02 c0 00 11 00 00   octets 48 à 63
0x40  3e c1 00 00 00 11 00 00 00 02 28 28 a7 b0 80 29   octets 64 à 79
0x50  ea fc 81 58 90 70                                 octets 80 à 85
```

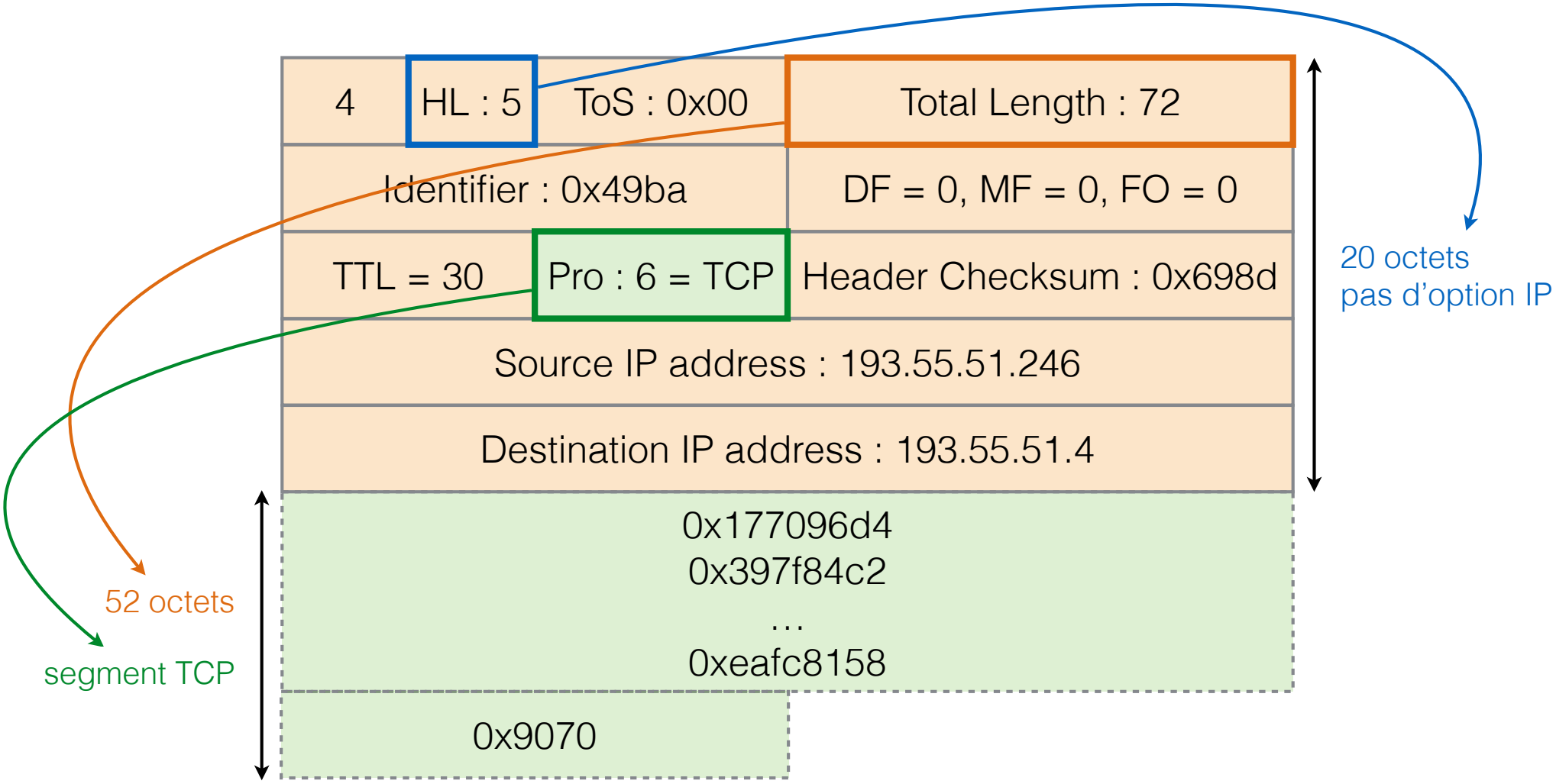
# Paquet IPv4 (sans options)



# Exemple de trace



# Exemple de trace



# Exemple de trace

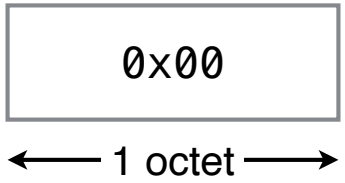
- Version : 0x4 → paquet IPv4
- Longueur de l'entête IP : 0x5 → 20 (5*4) octets
- ToS : 0x00
- Longueur totale : 0x0048 → 72 octets
- Identifiant : 0x49ba
- DF : 0, MF: 0, Fragment offset : 0 → pas de fragmentation
- TTL : 0x1e → 30 sauts possibles
- Protocole : 0x06 → TCP
- Somme de contrôle : 0x698d
- Adresse IP source : 0xc13733f6 → 193.55.51.246
- Adresse IP destination : 0xc1373304 → 193.55.51.4
- Longueur des données : 72 - 20 = 52 (longueur totale - longueur de l'entête)

# Options IP

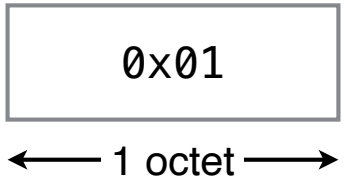
| Type | Option | Rôle |
|---|---|---|
| 0 | *End of Options List* | Utilisée pour aligner la fin des options et la fin de l'entête IP (octet de bourrage) |
| 1 | *No Operation* | Utilisée pour aligner les octets dans une liste d'options |
| 7 | *Record Route* (RR) | Utilisée pour enregistrer la route empruntée par le paquet IP |
| 68 | *Time Stamp* (TS) | Utilisée pour enregistrer le temps (en temps universel) où chaque équipement réseau reçoit le paquet pendant son trajet du point d'origine à sa destination |
| 131 | *Loose Routing* | Si utilisée, permet de spécifier une liste (incomplète) de routes que le paquet doit emprunter lors de son parcours de la source à la destination |
| 137 | *Strict Routing* | Si utilisée, permet de spécifier la liste exhaustive de routes que le paquet doit emprunter lors de son parcours de la source à la destination |

# Options IP

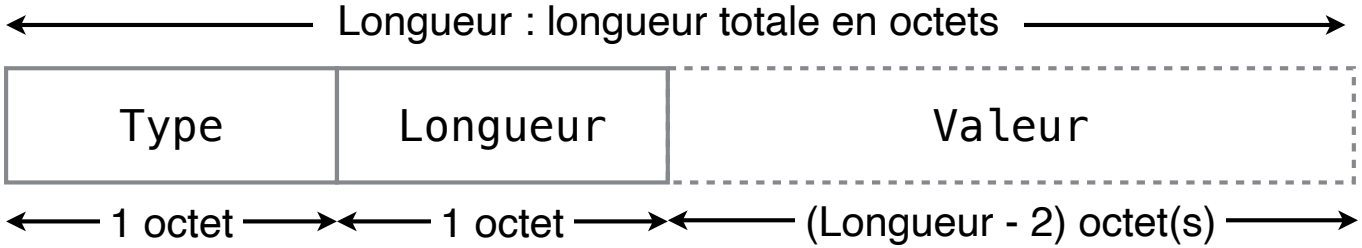
## Format de l'option *End of Options List* (EOL) : type = 0



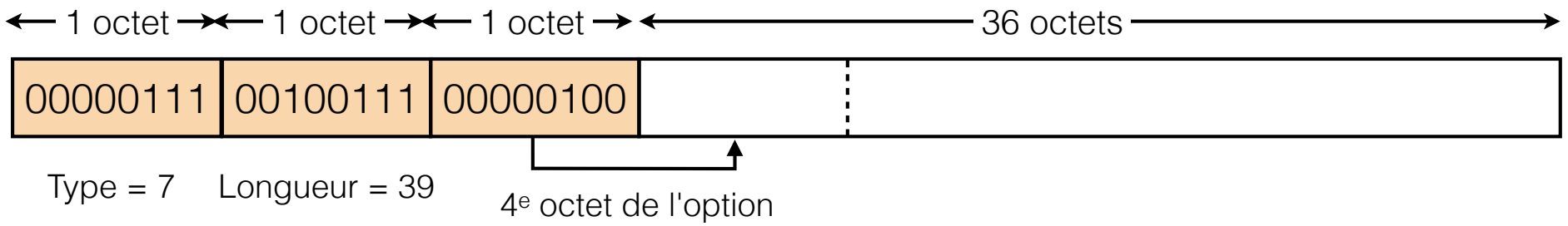

## Format de l'option *No OPeration* (NOP) : type = 1




## Format des options de type > 1

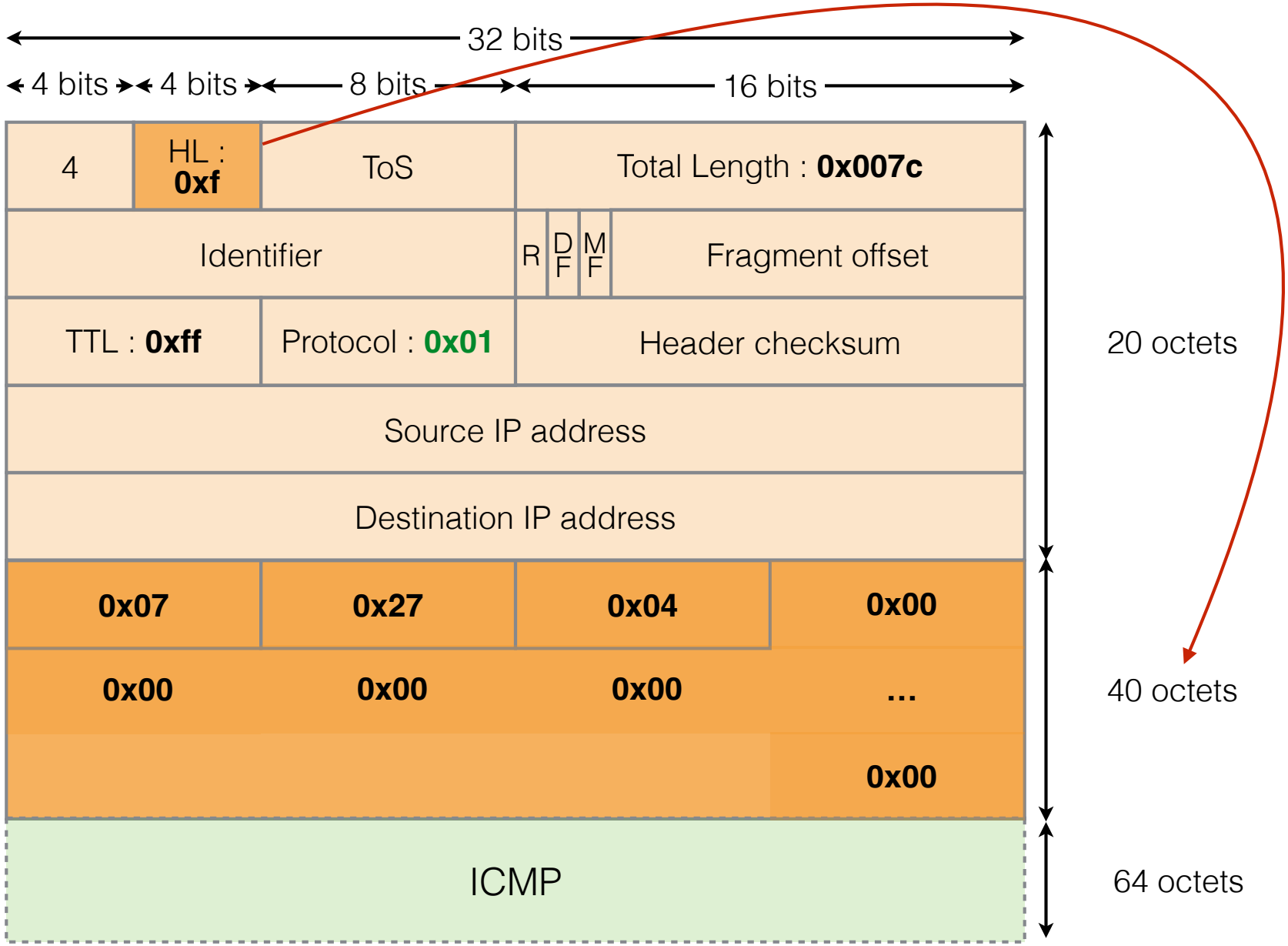


# Option Record Route

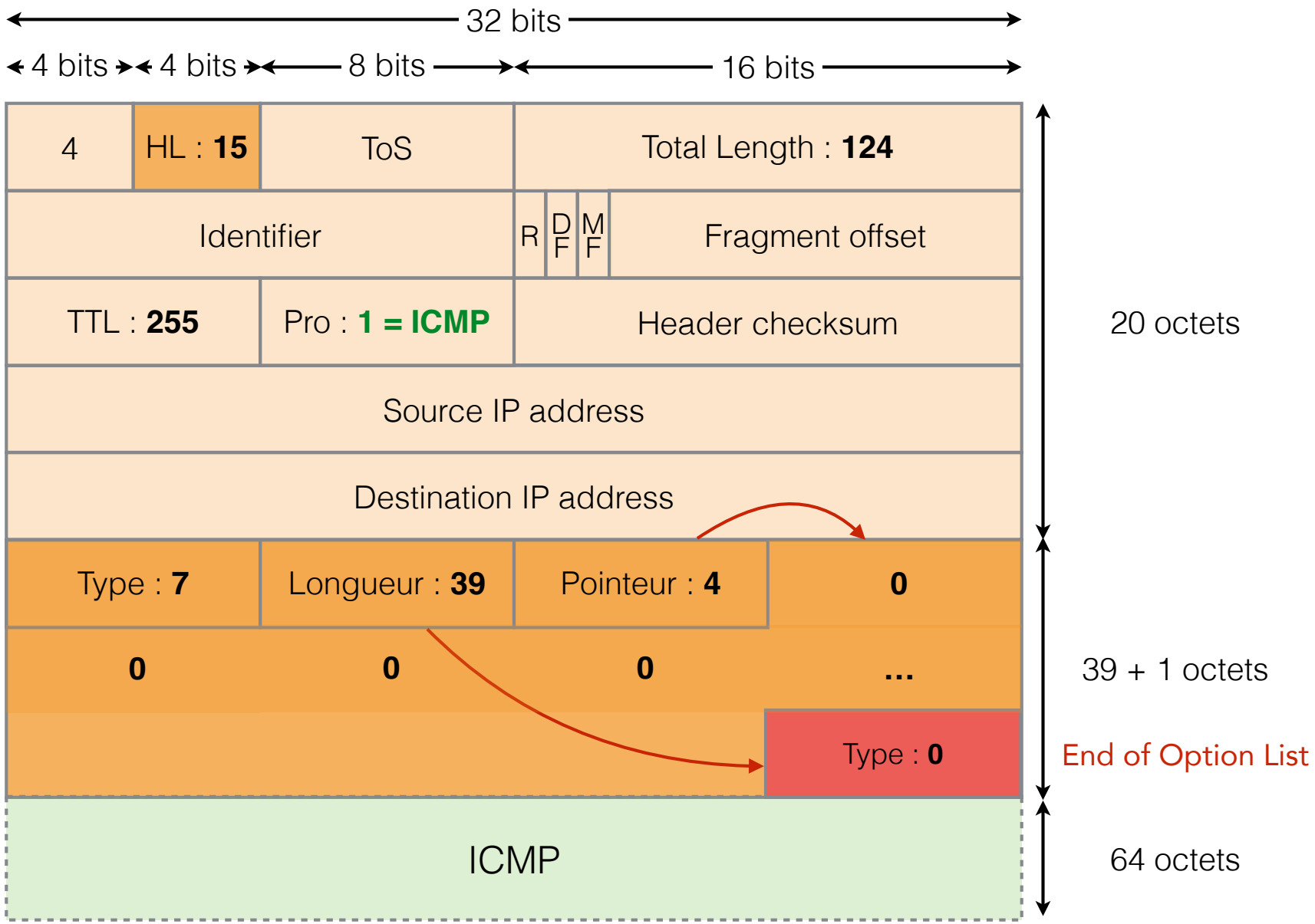



```
08 00 20 0a ac 96 08 00 20 0a 70 66 08 00|4f 00
00 7c cb c9 00 00 ff 01 b9 7f 84 e3 3d 05 c0 21
9f 06 07 27 04 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 08 00 a2 56 2f 00
00 00 29 36 8c 41 00 03 86 2b 08 09 0a 0b 0c 0d
0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d
1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d
2e 2f 30 31 32 33 34 35 36 37
```

# Paquet IPv4 avec option Record Route

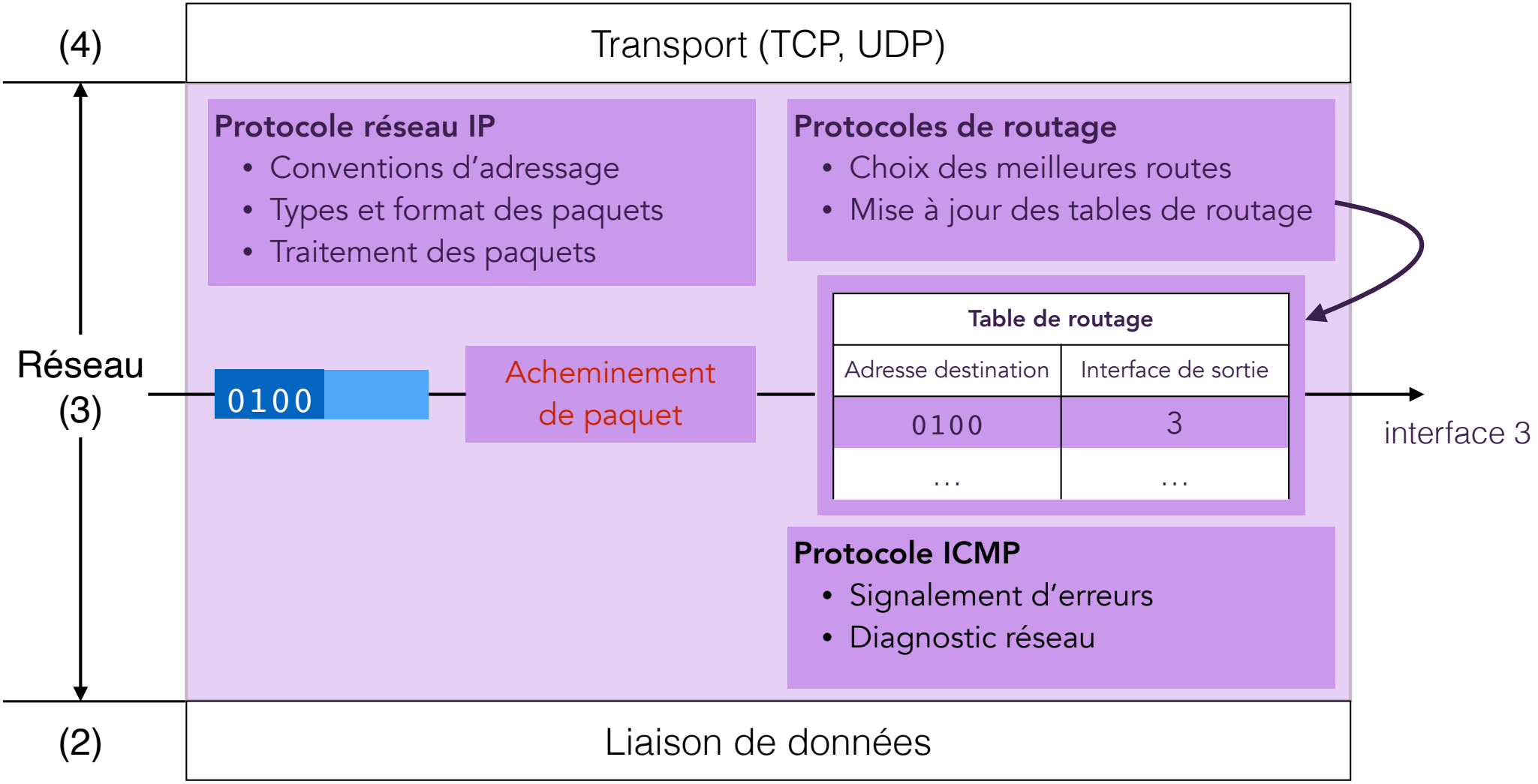


# Paquet IPv4 avec option Record Route



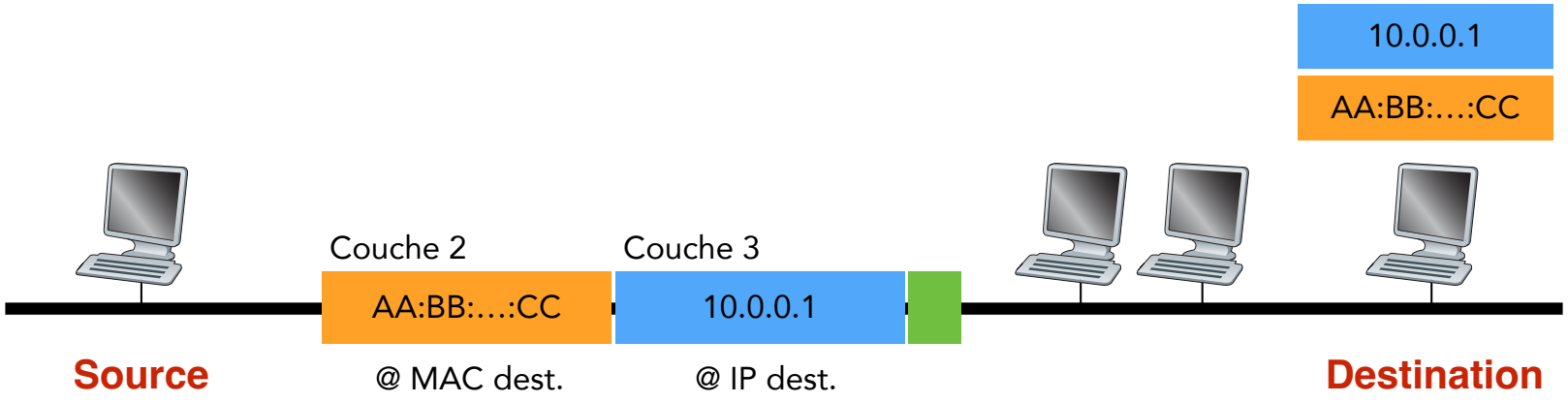
# Acheminement IP

# Couche réseau



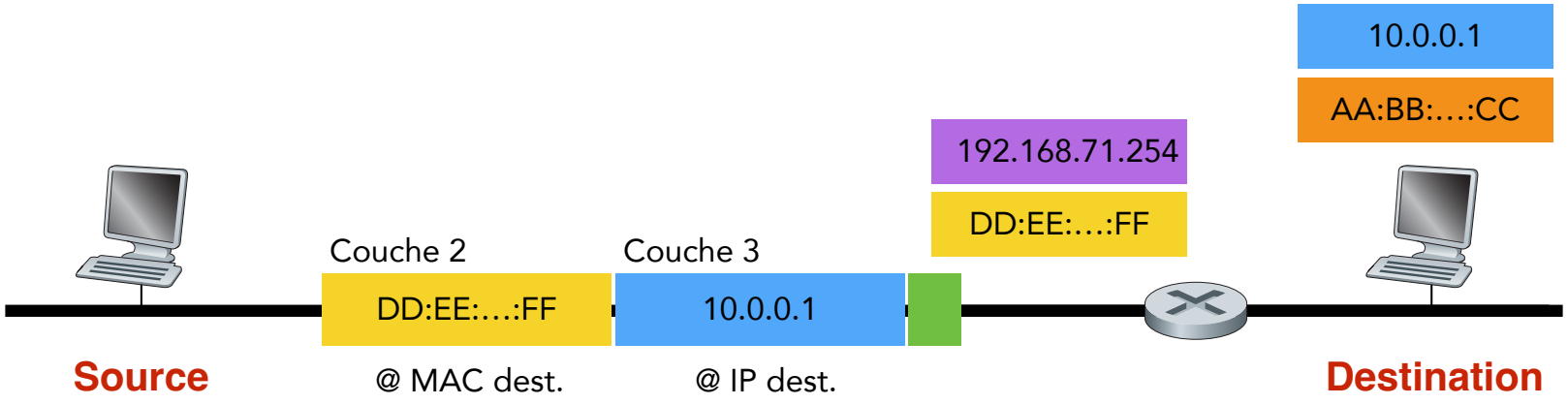
# Acheminement direct

- La machine de destination est sur le même sous-réseau que la machine source (pas de routeur entre la source et la destination)
  - La source envoie des paquets IP encapsulés dans des trames dont l'adresse MAC de destination et l'adresse IP de destination sont celles de la machine de destination



# Acheminement indirect

- La machine de destination n'est pas sur le même sous-réseau que la machine source (au moins un routeur les sépare)
  - La source envoie des paquets IP encapsulés dans des trames dont
    - l'adresse IP de destination est celle de la destination finale
    - l'adresse MAC de destination est celle du routeur de sortie du sous-réseau de la source (passerelle ou *gateway*)



# Table de routage IP

- Table de routage IP ou table d'acheminement IP ou FIB (*Forwarding Information Base*)

| Destination | Masque | Suivant | Interface |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

  - La colonne Destination indique la destination que permet de joindre cette entrée
    - adresse IP d'un réseau ou d'une machine (importante)
  - La colonne Masque (*Mask*) spécifie le masque associé à la destination
    - si la destination est une machine le masque est « 255.255.255.255 »
  - La colonne Suivant (*Gateway*) indique l'adresse IP du prochain routeur
    - en cas de routage direct, la colonne contient « * » ou « 0.0.0.0 »
  - La colonne Interface indique l'interface sur laquelle le paquet doit être transmis pour suivre la route considérée

# Acheminement des paquets

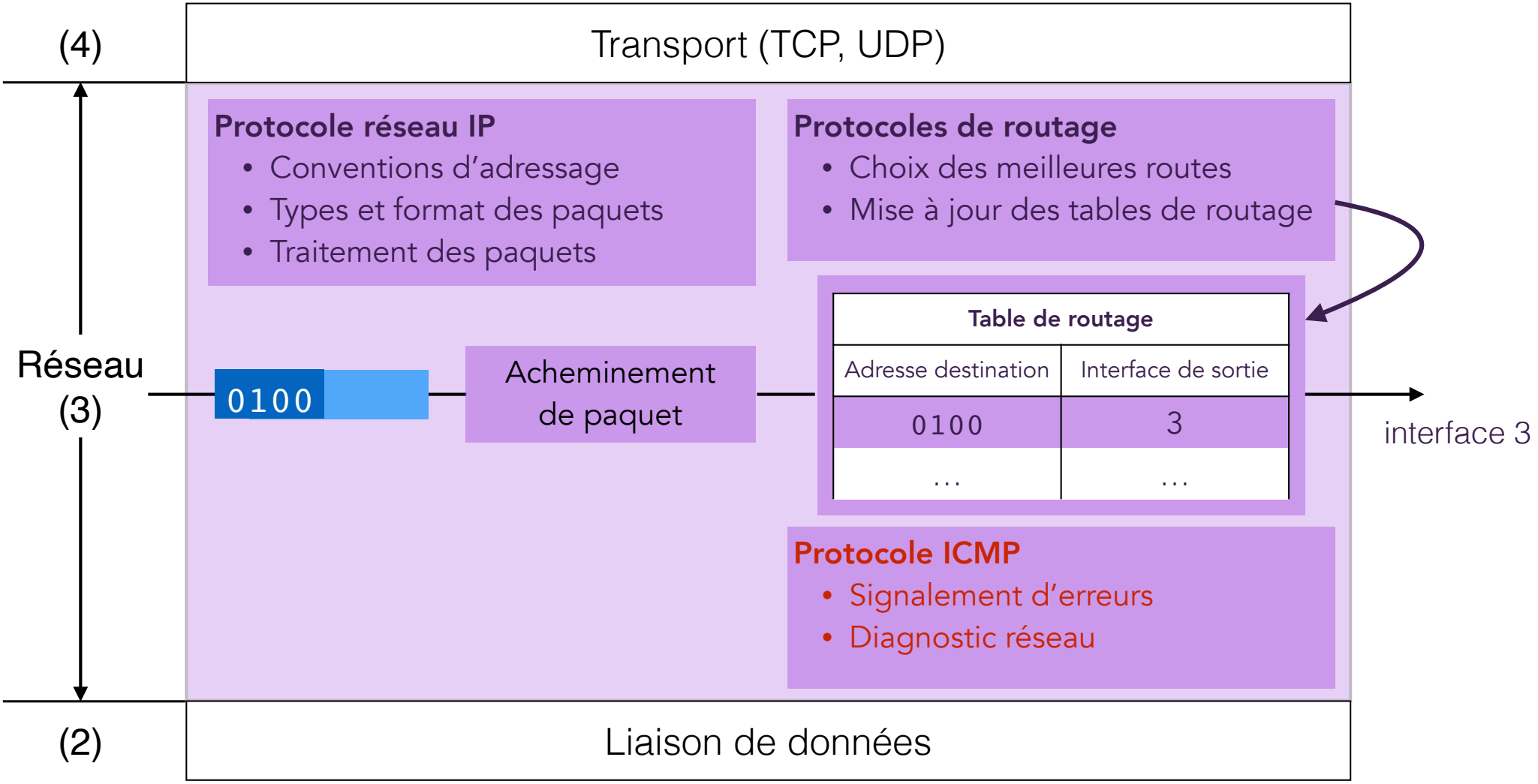
- Chaque machine (hôte ou routeur) maintient une table de routage
  - hôte : table simple généralement configurée manuellement
  - routeur : table complexe mise à jour à l'aide de protocoles de routage
- À la réception d'un paquet
  - la machine consulte l'adresse de destination du paquet
  - inspecte sa table de routage pour déterminer la « meilleure » entrée correspondant à cette adresse
  - achemine le paquet sur l'interface indiquée par cette entrée
- Exemple

| Destination | Masque | Suivant | Interface |
|---|---|---|---|
| 10.0.0.192 | 255.255.255.224 | 10.0.0.63 | eth0 |
| 10.0.0.0 | 255.255.255.192 | * | eth0 |
| 10.0.0.128 | 255.255.255.192 | * | eth1 |
| 0.0.0.0 | 0.0.0.0 | 10.0.0.191 | eth1 |

  - un paquet à destination de 10.0.0.136 est envoyé directement sur l'interface eth1
  - un paquet à destination de 10.0.0.200 est envoyé indirectement au routeur 10.0.63 sur l'interface eth0
  - un paquet à destination de 10.0.1.8 est envoyé indirectement au routeur 10.0.191 sur l'interface eth1

# Protocole ICMP

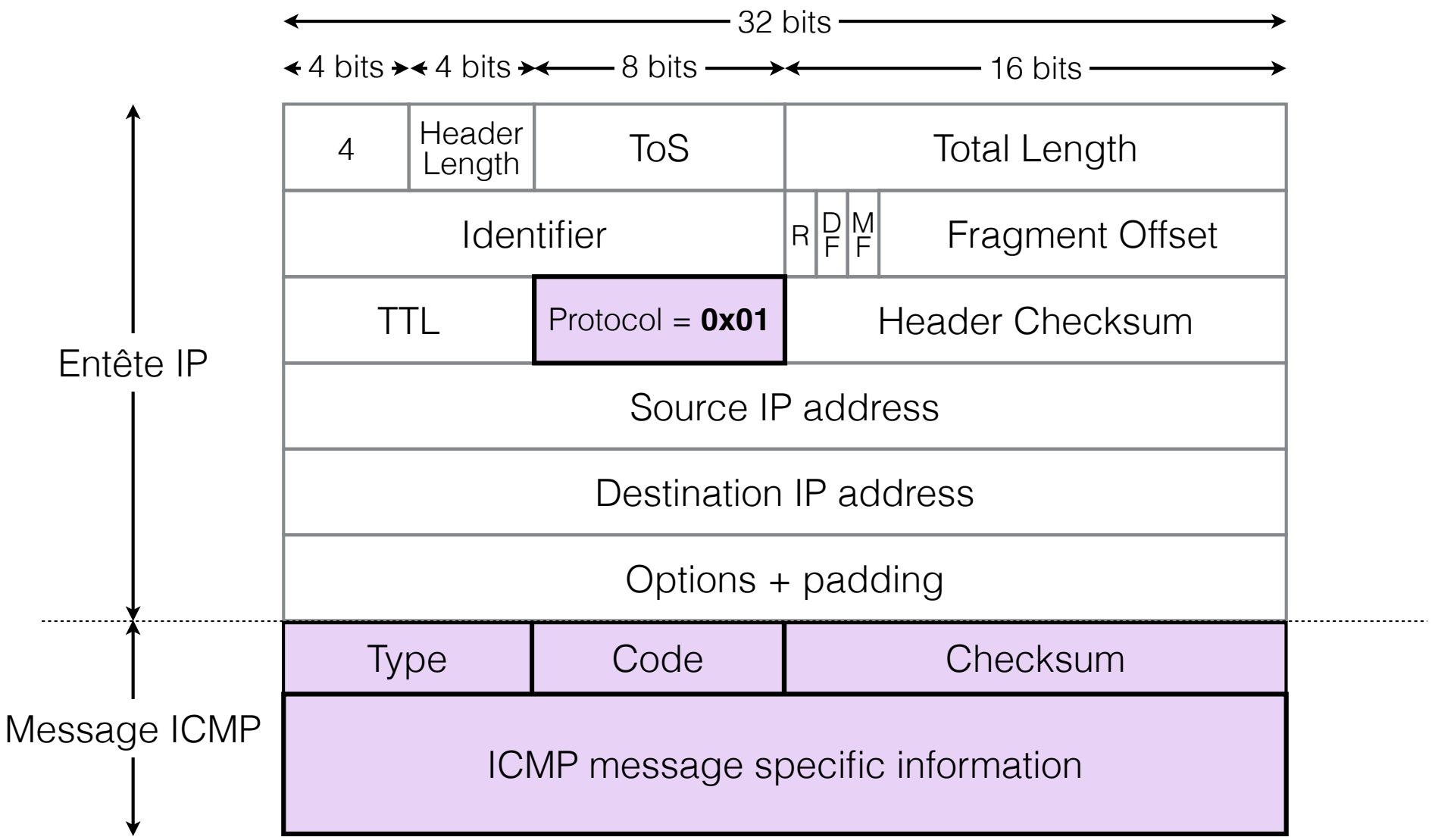
# Couche réseau



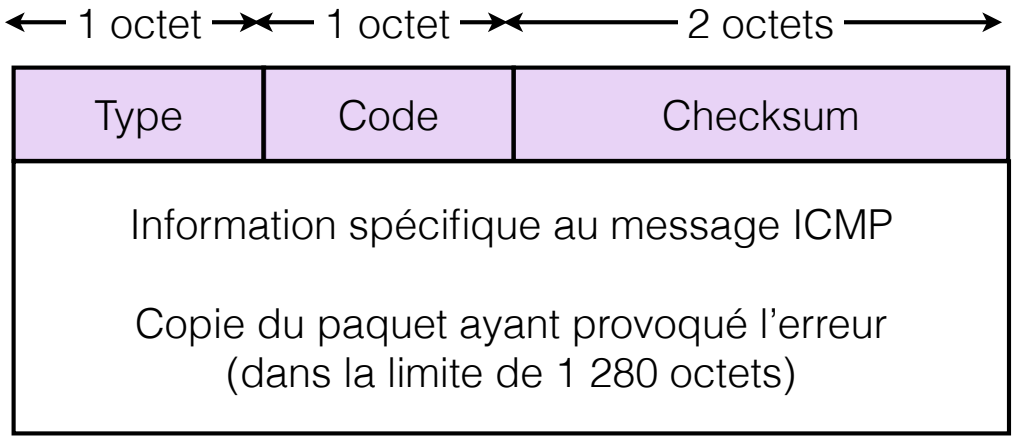
# Protocole ICMP

- *Internet Control Message Protocol*
- Fonctionnalités
  - diagnostics d'erreur en cas de problème d'acheminement ou de délivrance
    - temps de vie d'un paquet dépassé
    - paquet trop gros ne pouvant être fragmenté
    - destination inaccessible
    - …
  - tests de connectivité
    - pour vérifier si une machine est joignable
    - pour identifier des problèmes de routage
- Implémenté au dessus d'IP
  - message encapsulé dans un paquet IP
    - champ Protocol IP : 1
    - au même niveau que TCP (6) ou UDP (17)

# Encapsulation dans IP



# Message ICMP

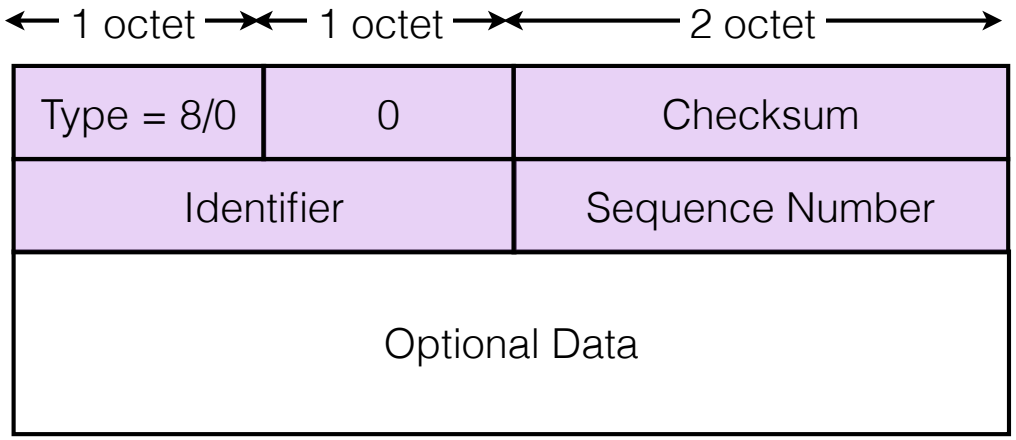



- Type : nature du message ICMP
  - messages d'erreur
  - messages de test
- Code : cause de l'erreur (en cas de message d'erreur)
- Checksum : somme de contrôle
  - Vérification de l'intégrité
    - du message ICMP
    - d'un pseudo-entête IP (similaire à celui de TCP et d'UDP)

# Types et codes ICMP

| Type | Code | Message |
|---|---|---|
| 0 | 0 | Echo Reply |
| 3 | 0 | Destination Network Unreachable |
| 3 | 1 | Destination Host Unreachable |
| 3 | 2 | Destination Protocol Unreachable |
| 3 | 3 | Destination Port Unreachable |
| 3 | 6 | Destination Network Unknown |
| 3 | 7 | Destination Host Unknown |
| 4 | 0 | Source Quench |
| 5 | 0 | Redirect |
| 8 | 0 | Echo Request |
| 11 | 0 | Time Exceeded |
| 11 | 1 | Reassembly Time Exceeded |
| 12 | | Parameter Problem |
| 13 | | Timestamp |
| 14 | | Timestamp Reply |
| 15 | | Information Request |
| 16 | | Information Reply |
| 17 | | Address Mask Request |
| 18 | | Address Mask Reply |

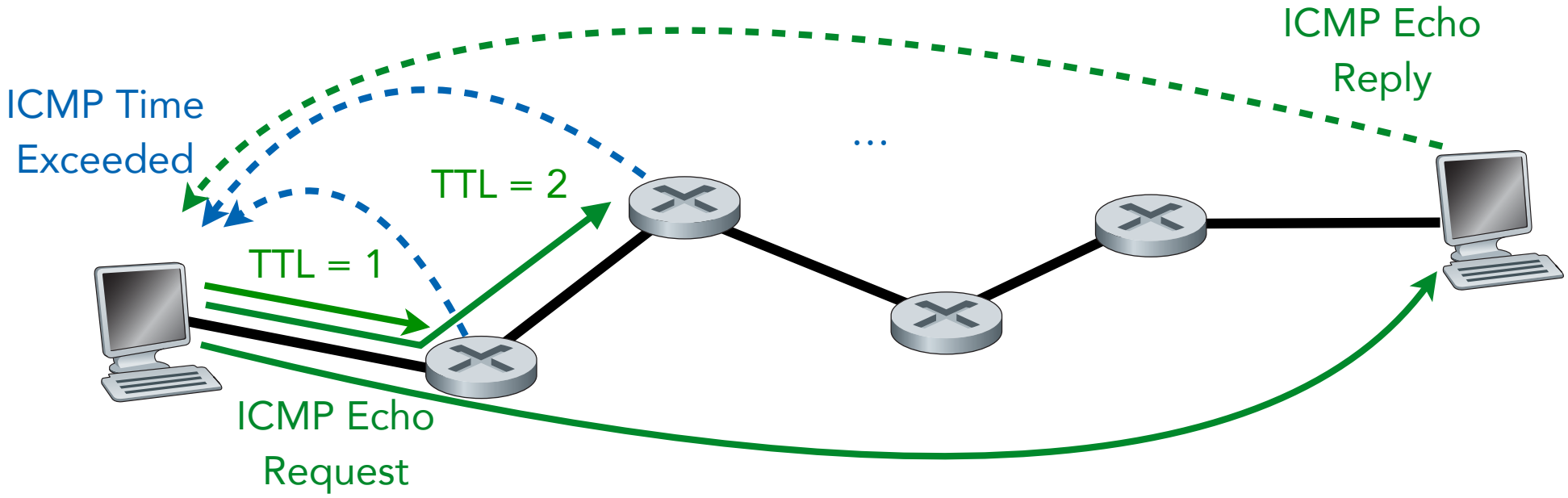
# Echo Request (Type 8) / Echo Reply (Type 0)




- Pour vérifier si une machine est joignable
  - une machine envoie un Echo Request à la machine dont elle veut tester l'accessibilité, celle-ci lui répond par un Echo Reply
  - envoyer plusieurs Echo request à une même machine permet de faire des statistiques (délai moyen AR, paquets perdus)
- Champ Identifier
  - permet de faire correspondre les messages Echo Reply reçus aux messages Echo Request envoyés (si envoyés à différentes machines)
- Champ Sequence Number
  - permet de faire correspondre un Echo Reply à l'Echo request correspondant (si plusieurs Echo Request envoyés à la même machine)
- Exploités par la commande Unix « ping »
  - Ex : ping -c 3 10.0.0.1

# Command Traceroute

- La commande Unix Traceroute permet à une machine source de connaître la route complète vers une destination
- Envoi d'une succession de messages ICMP Echo Request en incrémentant le TTL du paquet IP qui les encapsule



# Conclusion

- Champs d'entête du paquet IP
  - taille comprise entre 20 et 60 octets
  - les erreurs sur l'entête sont détectées par le champ *Header Checksum*
  - la durée de vie du paquet est limitée par le champ TTL
- Longueur d'un paquet IP
  - les paquets trop longs peuvent être
    - fragmentés
    - détruits
- Charge utile du paquet IP
  - identifié par le champ Protocole
    - 6 : TCP
    - 17 : UDP
    - 1 : ICMP
- Acheminement IP
  - direct ou indirect
  - réalisé par consultation des tables d'acheminement
- Protocole ICMP
  - pour diagnostiquer des erreurs de routage ou de livraison
  - pour tester la connectivité
    - commandes ping et traceroute

# A faire

- Cours 6
  - à relire attentivement
- Devoir 6 sur Moodle
  - date de rendu : dimanche 13 octobre