La pile TCP/IP

Adressage IP, ARP, ICMP, Routage, TCP, UDP, service Internet

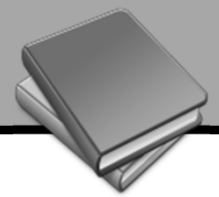
Christian Bulfone
christian.bulfone@gipsa-lab.fr
www.gipsa-lab.fr/~christian.bulfone/MIASHS-DCISS

Master MIASHS/DCISS
Année 2024/2025

Plan du cours

- Un peu d'histoire ...
- Sources d'information
- Adressage IP
- Les protocoles ARP, ICMP
- La couche Transport : UDP et TCP
- Quelques services : DNS, FTP, Telnet, SSH, DHCP
- Le routage

Sources d'information



- RFC : *Request For Comments*
  - tous les standards Internet
  - source (via ftp) sur nic.ddn.mil (copie à ftp.inria.fr)
- FYI : *For Your Information*
- W. Richard Stevens - TCP/IP illustré
  - Volume 1 : Les protocoles
  - Volume 2 : La mise en œuvre
  - Volume 3 : Les nouveaux protocoles

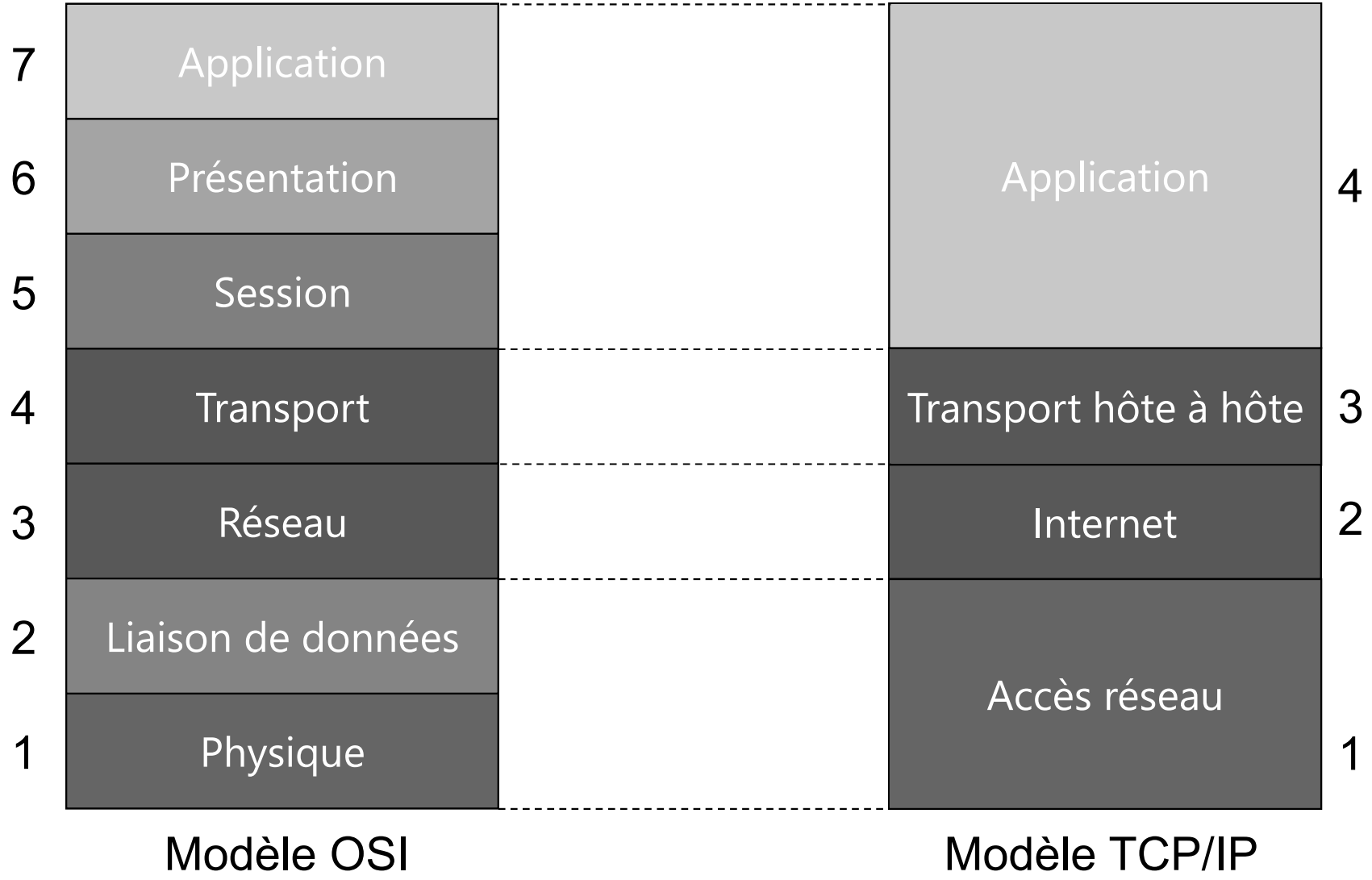
Historique TCP/IP

- Fruits des recherches du DARPA
- 1969 : ARPAnet
  - favorise les réseaux résistants aux destructions partielles
  - grand succès
- 1978 : stade opérationnel
  - gestion du réseau confiée à la DISA (ex-DCA)
- 1983 : les protocoles TCP/IP deviennent des standards militaires
- Intégration de TCP/IP en standard sous Unix BSD
  - entrée dans le monde universitaire
  - développement d'applications réseaux avec les sockets
- 1990 : explosion d'IP en Europe (non académique)

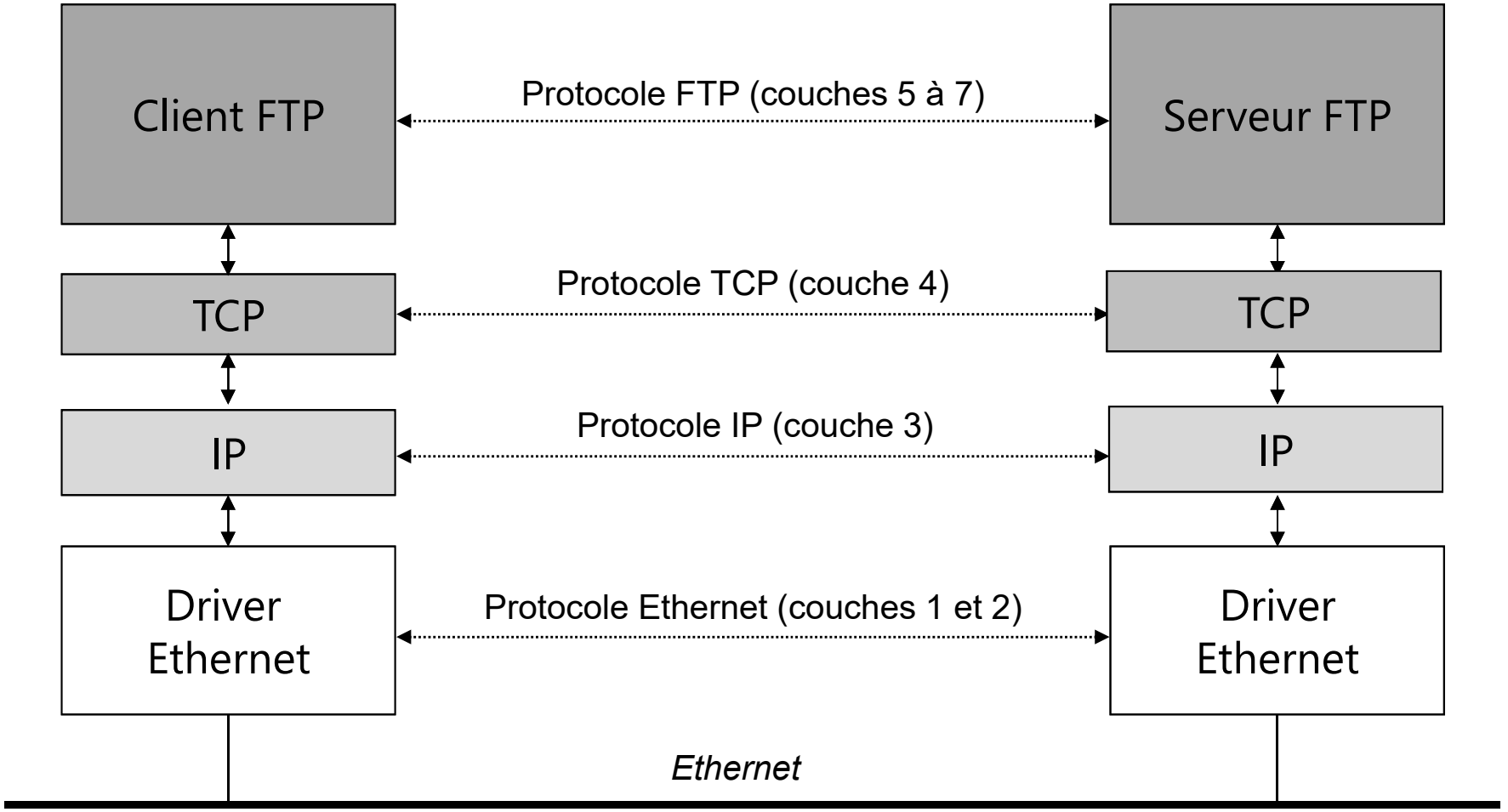
Historique TCP/IP

- 3 facteurs principaux expliquent la montée en puissance de TCP/IP depuis 1990
  - l'interopérabilité : un protocole commun sur des produits provenant de différents constructeurs
  - l'intérêt commercial : l'Internet est basé sur les protocoles et services TCP/IP
  - l'augmentation du nombre d'outils de gestion de réseau (SNMP - *Simple Network Management Protocol* - notamment)
- Historiquement créé à la demande du ministère de la Défense des Etats-Unis, on désigne TCP/IP souvent sous le nom de modèle DoD (*Department Of Defense*)
- Les protocoles TCP/IP, comme ceux du modèle OSI, étant organisés en couches, on emploie aussi le terme de *pile* TCP/IP

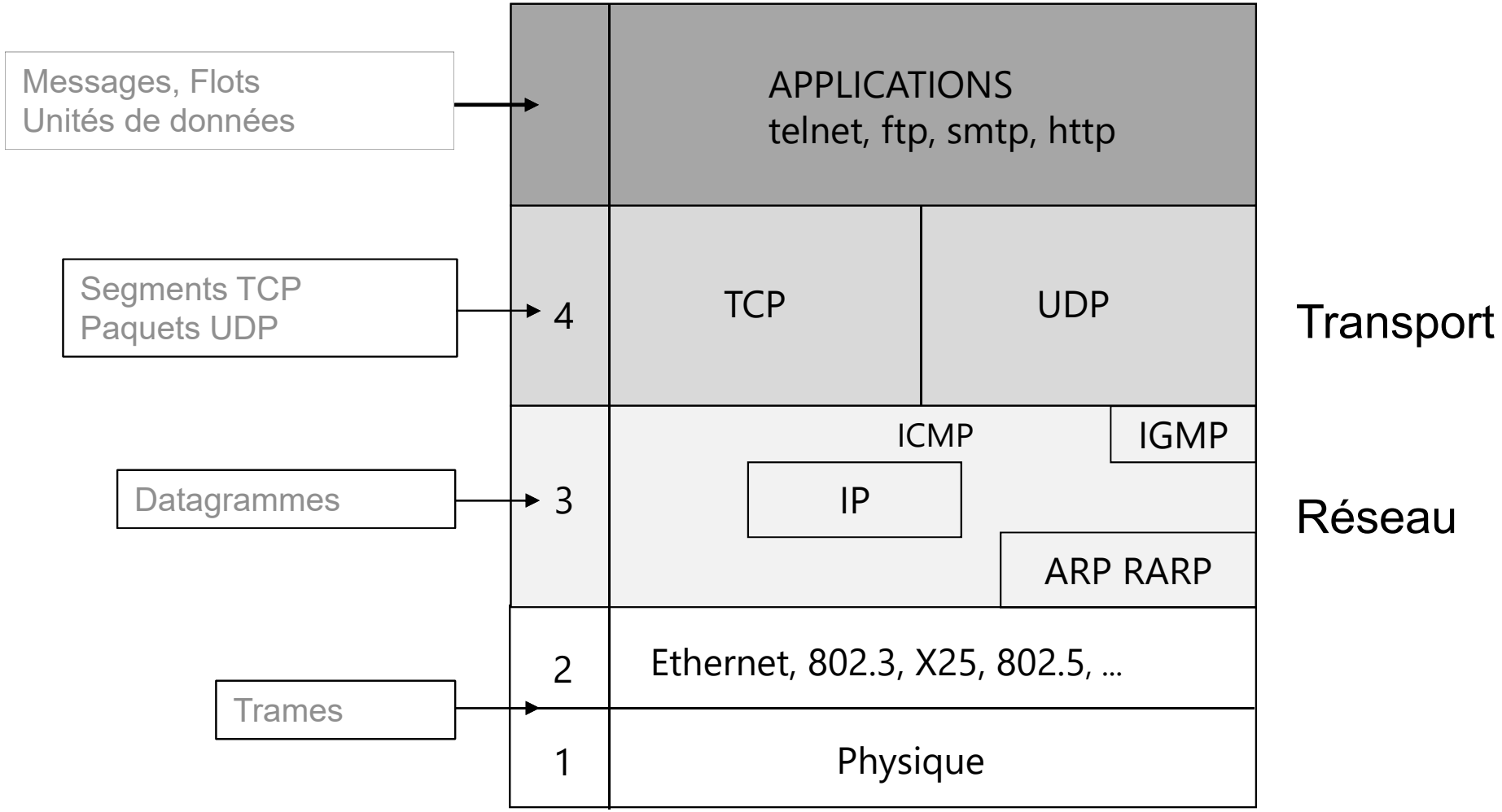
Modèles OSI / TCP/IP



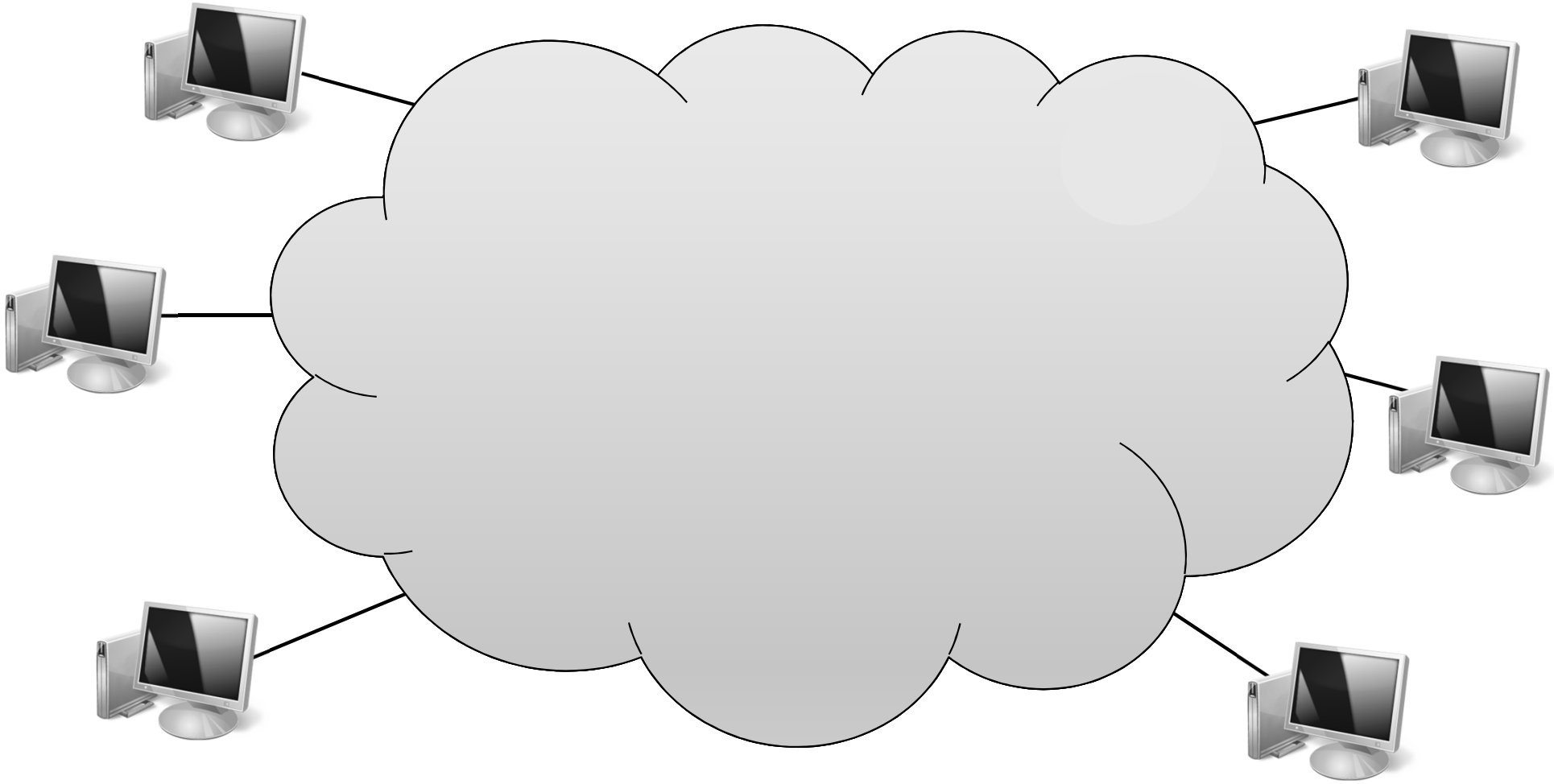
Organisation en couches



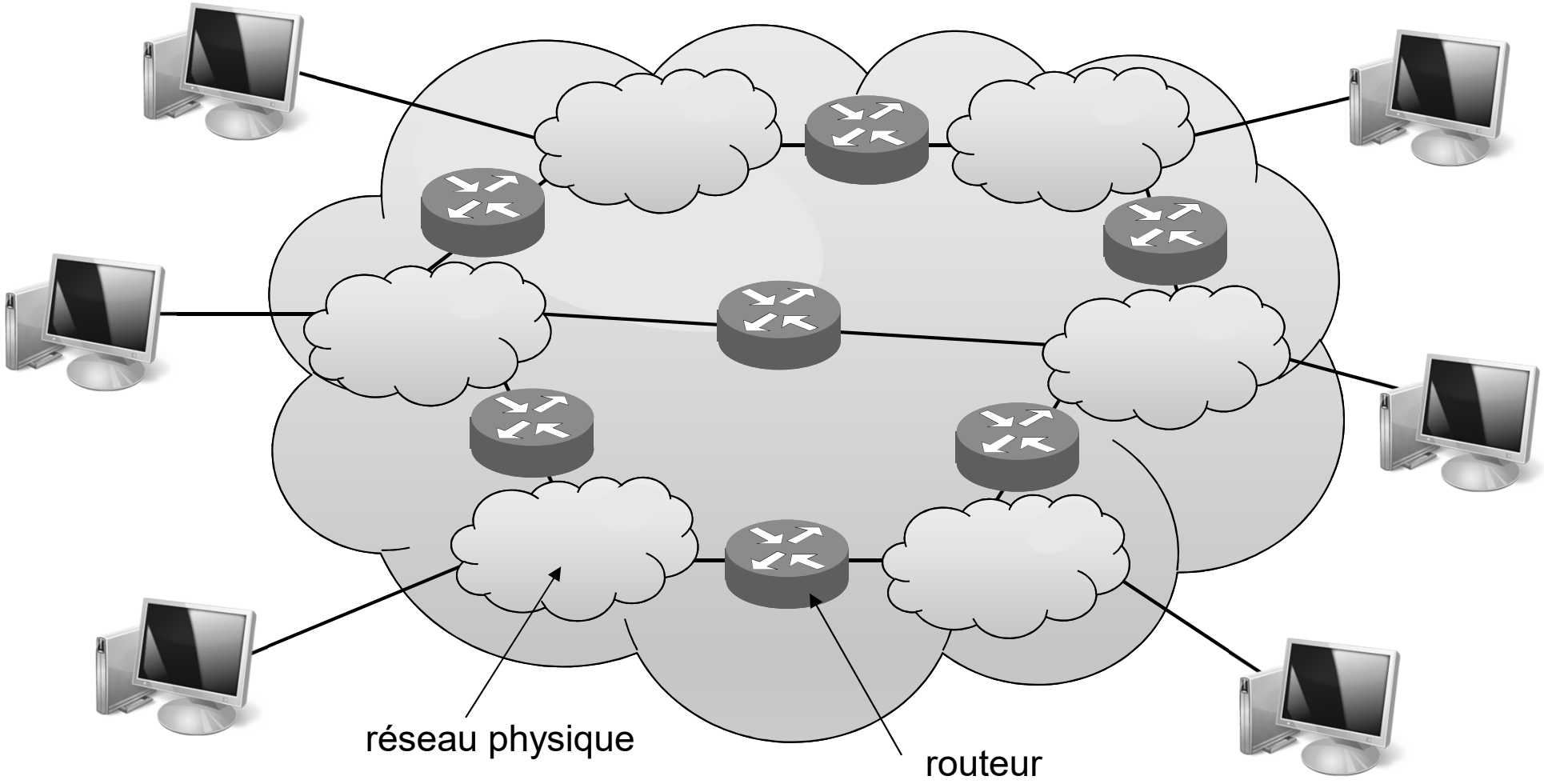
Organisation en couches



Internet vu par l'utilisateur



Internet : la réalité



Le protocole IP

Deux versions du protocole

- IPv4
  - Version historique encore répandue
  - Codage des adresses sur 32 bits
- IPv6
  - Déploiement toujours en cours depuis 2003 (en Europe et Etats-Unis)
  - Codage des adresses sur 128 bits
  - Cohabite avec la version 4
  - Nécessite une nouvelle pile de protocoles
  - Implémenté sur la plupart des OS

Apports IPv6

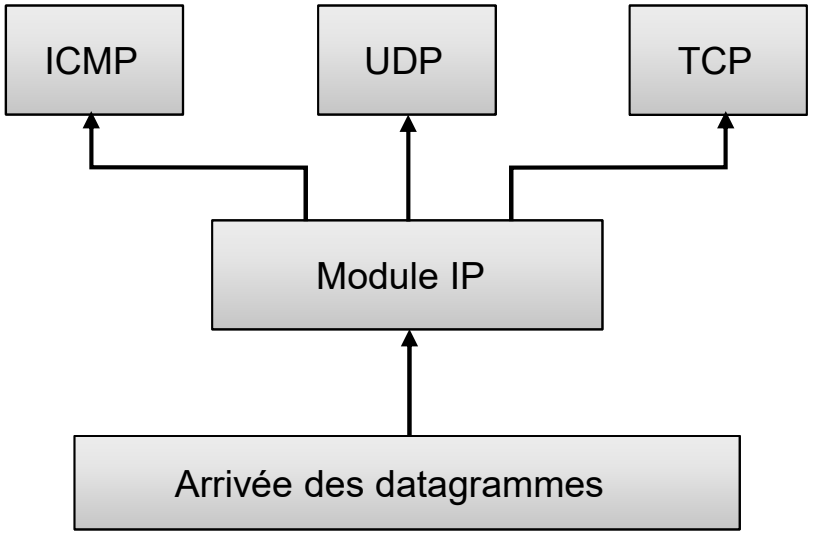
- Supporte des milliards d'ordinateurs, sans l'inefficacité de l'espace des adresses IP actuelles
- Réduit la taille des tables de routage
- Simplifie le protocole, pour permettre aux routeurs de router les datagrammes plus rapidement
- Fournit une meilleure sécurité (authentification et confidentialité)
- Accorde plus d'attention au type de service, et notamment aux services associés au trafic temps réel
- Facilite la diffusion multi-destinataire
- Donne la possibilité à un ordinateur de se déplacer sans changer son adresse
- Est évolutif

Fonctions d'IP

- Transporter des **datagrammes** de bout en bout
- Il faut connaître l'adresse IP d'un équipement pour communiquer avec lui
- Mode **sans connexion**
  - chaque datagramme est traité indépendamment des autres
- Pas de garantie de remise des datagrammes (non fiable ou *unreliable*)
  - stratégie de type « Best Effort » (au mieux)
- Assure le **routage**
- Peut fragmenter les messages

Fonctions d'IP

- Le démultiplexage



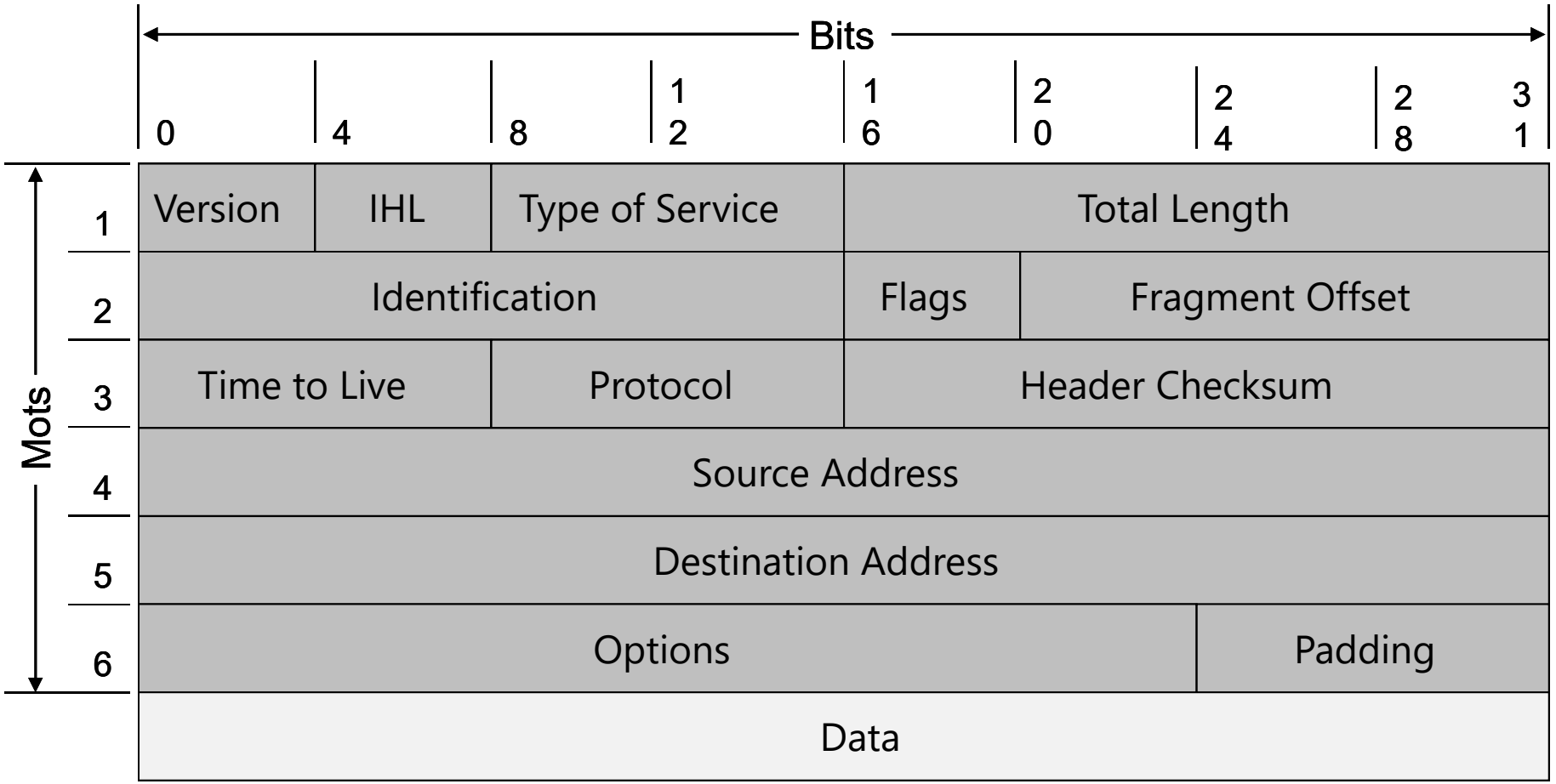
- Ce qu'IP ne fait pas :
  - le multiplexage
  - la vérification du séquencement
  - la détection de pertes
  - la retransmission en cas d'erreur,
  - le contrôle de flux

Format du datagramme IP (v4)

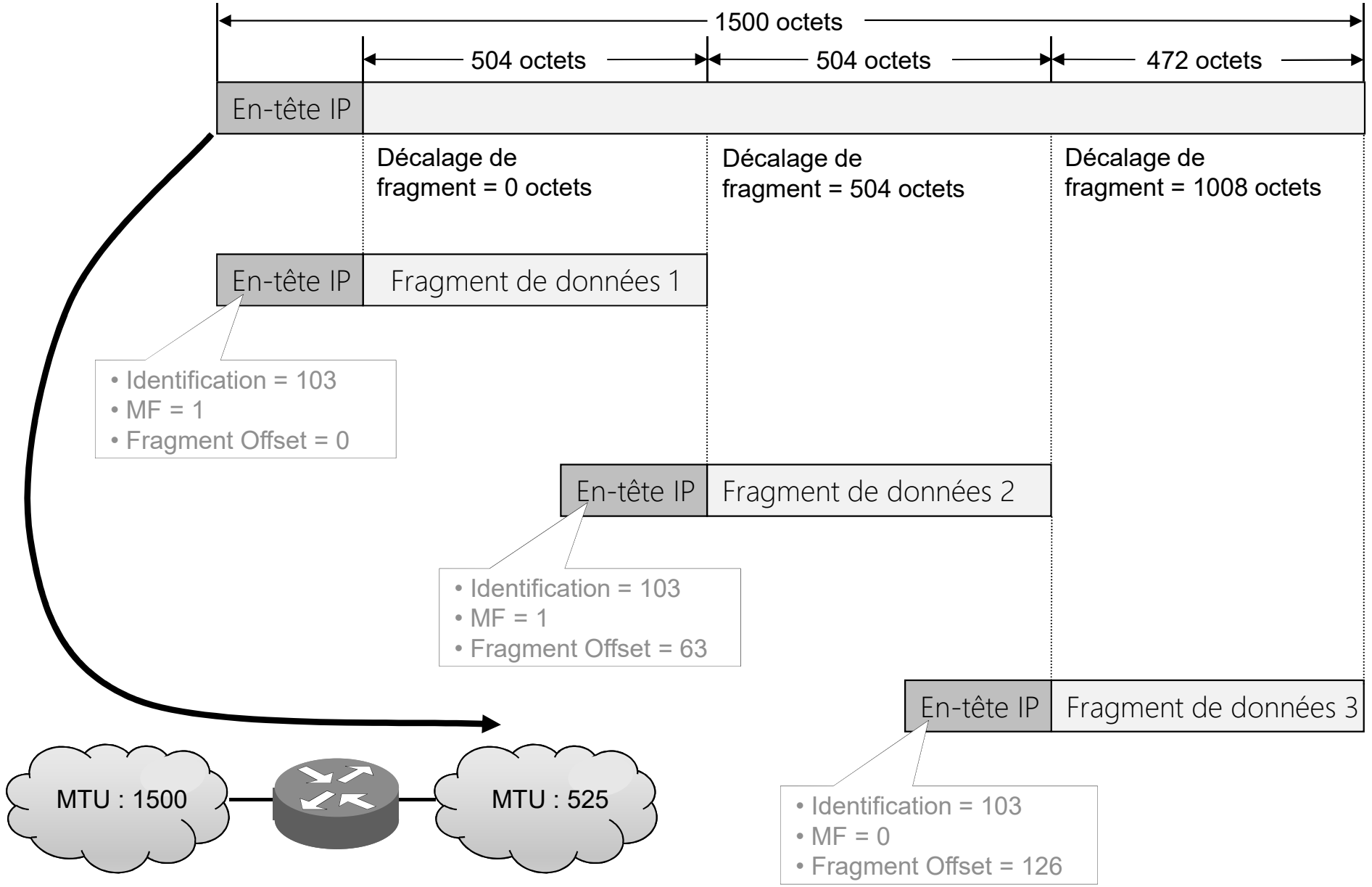
- Lors de l'émission, les données sont découpées en petits paquets, appelés **datagrammes** IP
- Les datagrammes sont tous composés :
  - d'un en-tête
  - suivi d'une zone de données

- L'en-tête contient les adresses de l'**émetteur** et du **destinataire**
- Le routage est basé sur **l'adresse du destinataire**

Format du datagramme IP (v4)

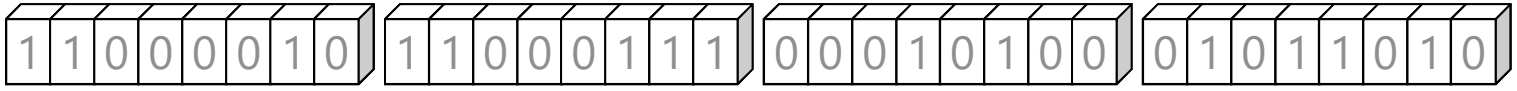


Fragmentation



La structure de l'adresse IP (v4)

- Chaque interface réseau d'un appareil possède une adresse IP unique au monde
  - Configurable par logiciel
  - Attribuée par le NIC (*Network Information Center*)
  - Codée sur 32 bits en notation décimale pointée
    - exemple : 194.199.20.90



Structure de l'adresse IP (v4)

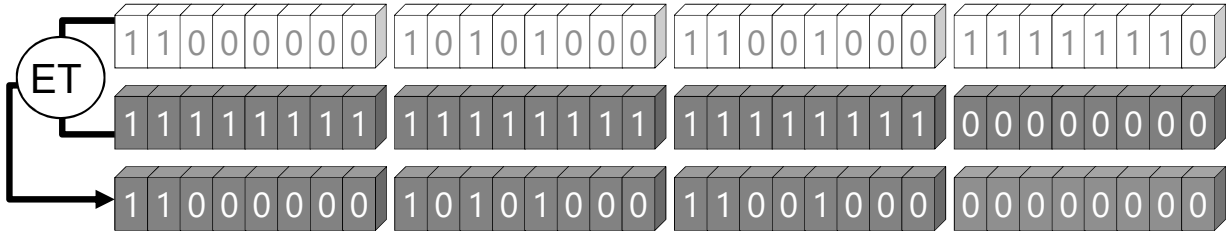
- Adresse hiérarchique
  - Une relation existe entre les adresses d'équipements voisins
- Structurée en deux parties :
  - le **préfixe**, donnant le numéro de réseau : ID de réseau ou « netid »
  - le **suffixe**, donnant le numéro de la machine (hôte) dans ce réseau : ID de station ou « hostid »
- Un **masque** (*netmask*) est associé à cette adresse
  - il permet au logiciel IP de déterminer le préfixe de réseau d'une adresse en calculant un ET logique avec le masque

Interface : eth0

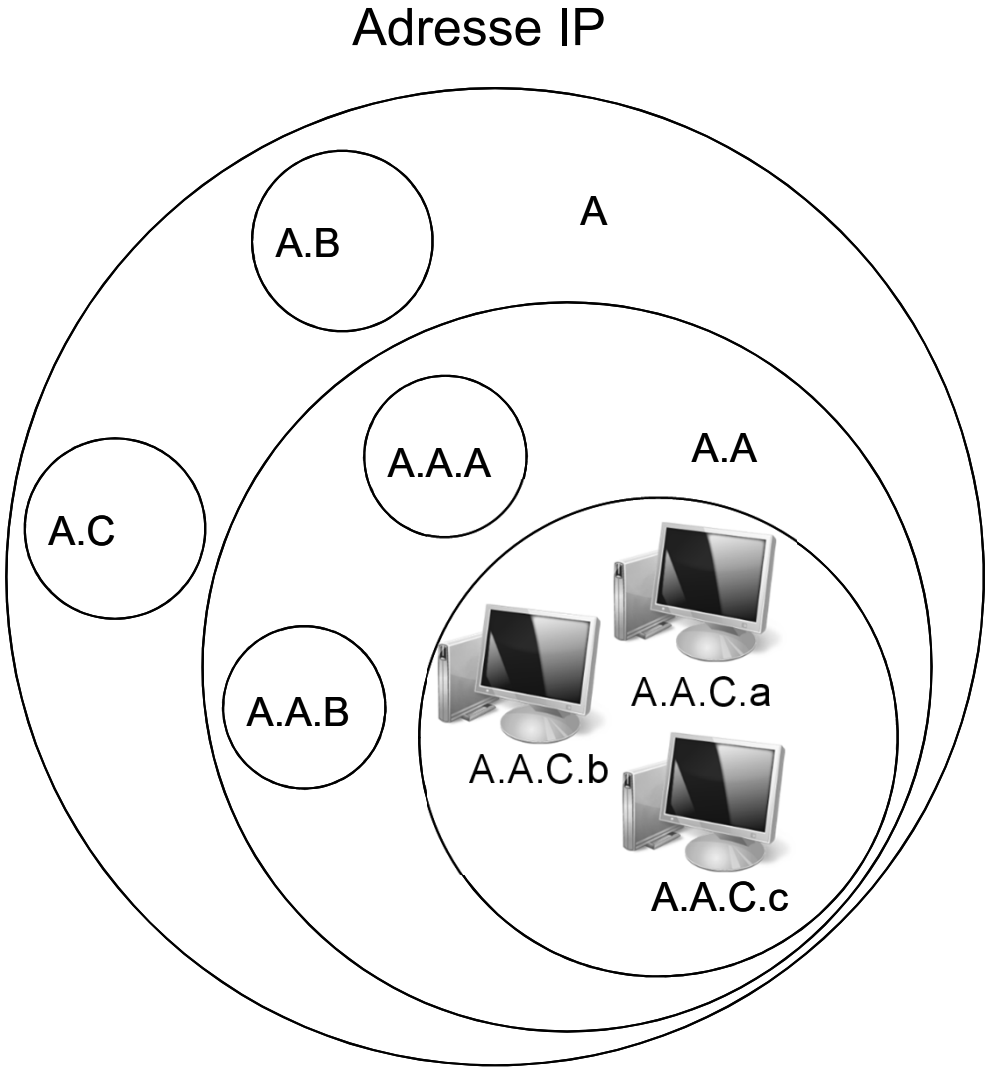
Adresse IP : 192.168.200.254

Masque réseau : 255.255.255.0

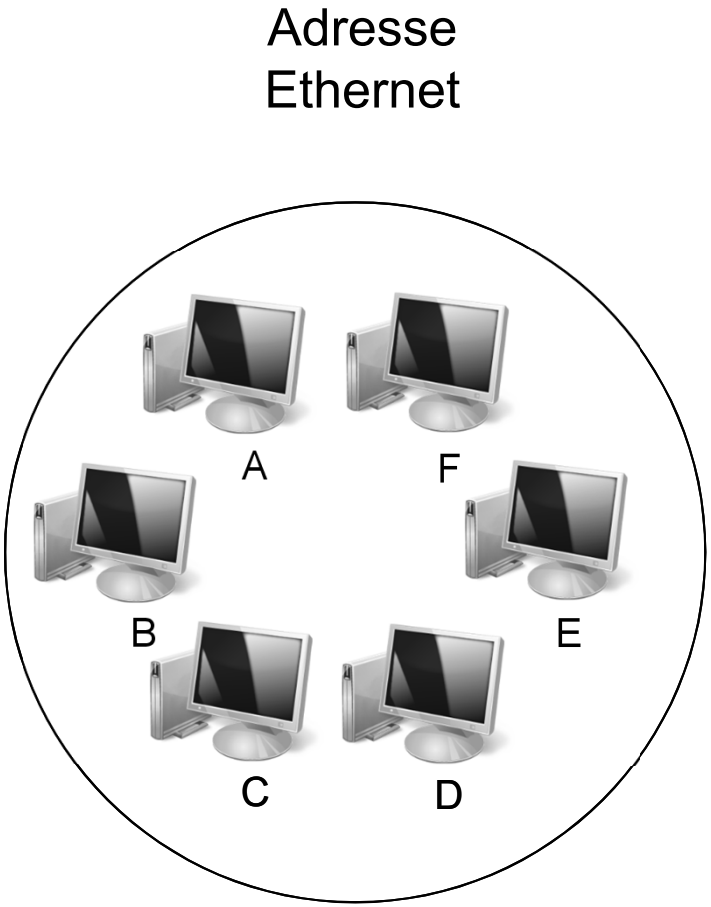
Préfixe réseau : 192.168.200.0



Espaces d'adressage (IPv4)



Espace d'adressage hiérarchique

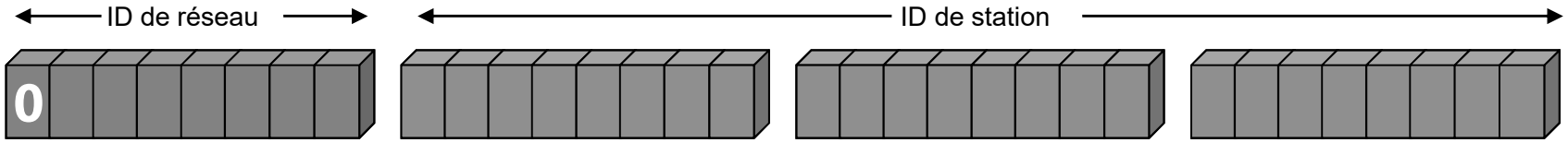


Espace d'adressage plat

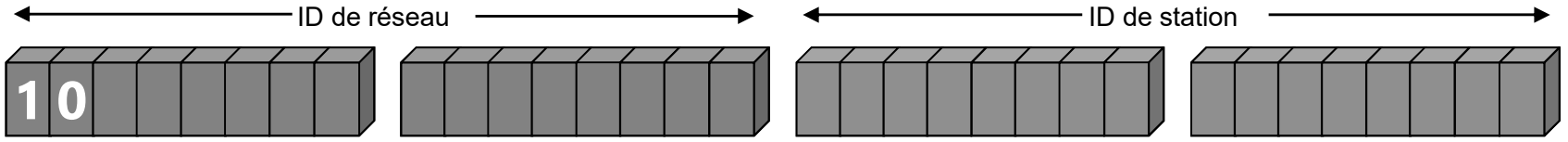
Notation de l'adresse IP (v4)

- Notation « classique »
  - Adresse IP / masque en décimal pointé
  - 192.168.200.254/255.255.255.0
- Notation condensée
  - Adresse IP / nombre de bits à 1 du masque
  - 192.168.200.254/24

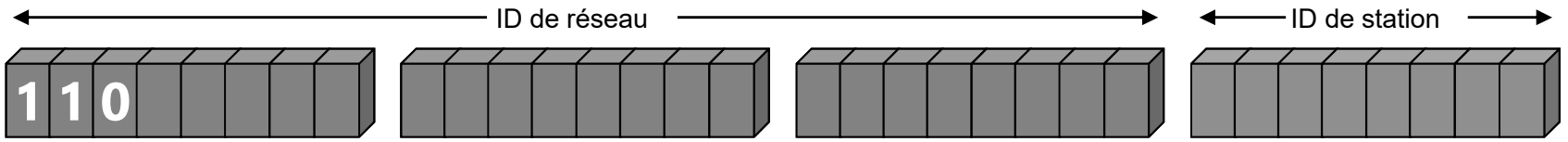
Structure de l'adresse IP (v4) Classes de réseaux



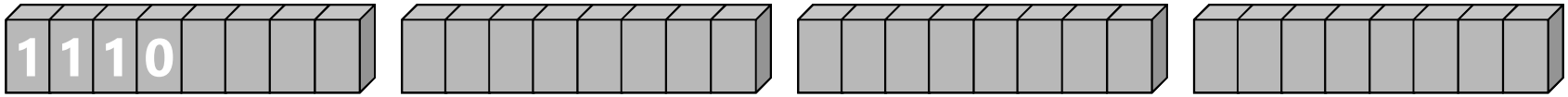
Classe A : de 0.0.0.0 à 127.255.255.255



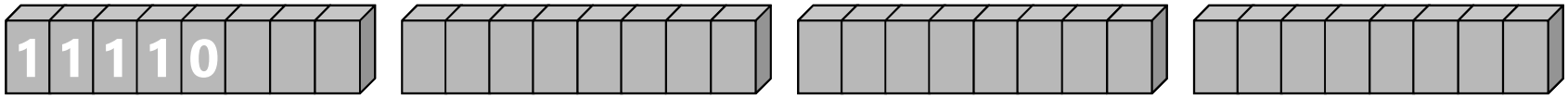
Classe B : de 128.0.0.0 à 191.255.255.255



Classe C : de 192.0.0.0 à 223.255.255.255



Classe D : de 224.0.0.0 à 239.255.255.255



Classe E : de 240.0.0.0 à 247.255.255.255

Structure de l'adresse IP (v4) Adresses spéciales

- Adresse de **réseau** :
  - Identificateur de réseau suivi de bits à 0
  - Exemples :
    - 125.0.0.0 = réseau 125 de classe A
    - 129.15.0.0 = réseau 129.15 de classe B
    - 192.168.30.0 = réseau 192.168.30 de classe C
- Adresse de **diffusion** ou *broadcast* :
  - Identificateur de réseau suivi de bits à 1
  - Exemples :
    - 125.255.255.255 = diffusion sur le réseau 125 de classe A
    - 129.15.255.255 = diffusion sur le réseau 129.15 de classe B
    - 192.168.30.255 = diffusion sur le réseau 192.168.30 de classe C

Structure de l'adresse IP (v4) Adresses spéciales

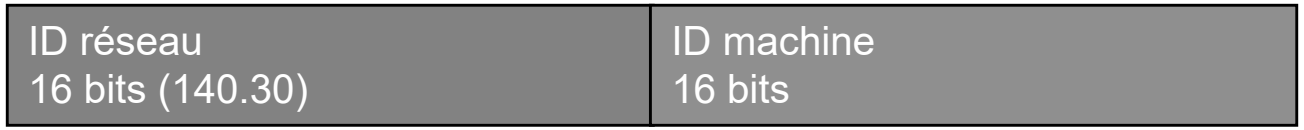
- Adresse de **machine** ou d'**hôte**
  - Exemples :
    - 125.5.6.7 = machine 5.6.7 du réseau 125 de classe A
    - 129.15.106.213 = machine 106.213 du réseau 129.15 de classe B
    - 192.168.30.11 = machine 11 du réseau 192.168.30 de classe C
- 127.x.x.x
  - adresse de **bouclage** (*loopback localhost*)
  - désigne la machine locale
- 0.0.0.0
  - utilisé quand une machine ne connaît pas son adresse
  - utilisé pour désigner la route par défaut dans la table de routage

Sous-réseaux IP (v4)

- Découpage d'un réseau en entités plus petites
  - sous-réseau ou “*subnet*”
  - permet meilleure structuration du réseau du site
  - décidé par l'administrateur du site
  - adresse de sous-réseaux prélevé sur la partie *host-id*
  - longueur comptée en bits décidée par l’administrateur
  - tous les équipements réseaux doivent utiliser la notion de sous-réseau (stations, serveurs de terminaux, routeurs, imprimantes…)
  - interconnexion des sous-réseaux impérativement par des routeurs

Adressage de sous-réseaux

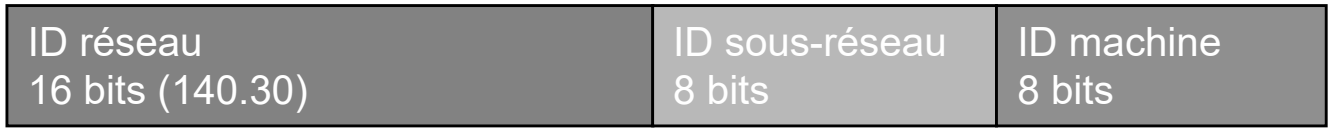
- Exemple : réseau de classe B 140.30.0.0



  - masque de réseau par défaut 255.255.0.0 si aucun sous-réseau n'est défini

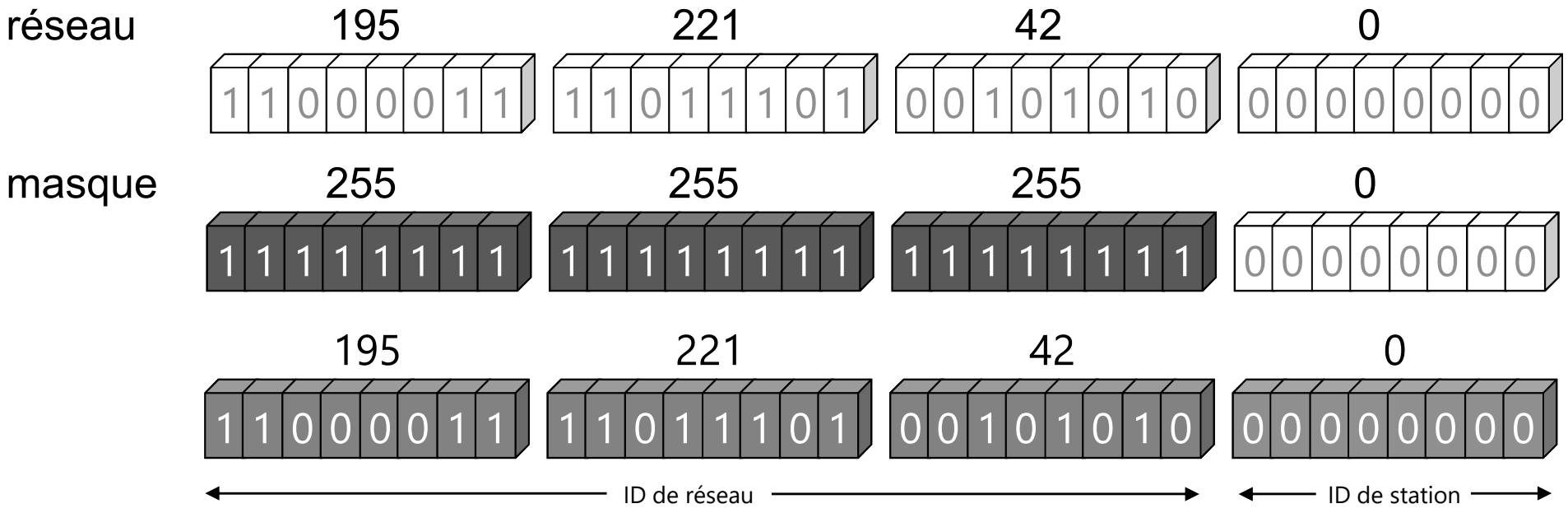
Adressage de sous-réseaux

- Exemple : réseau de classe B 140.30.0.0

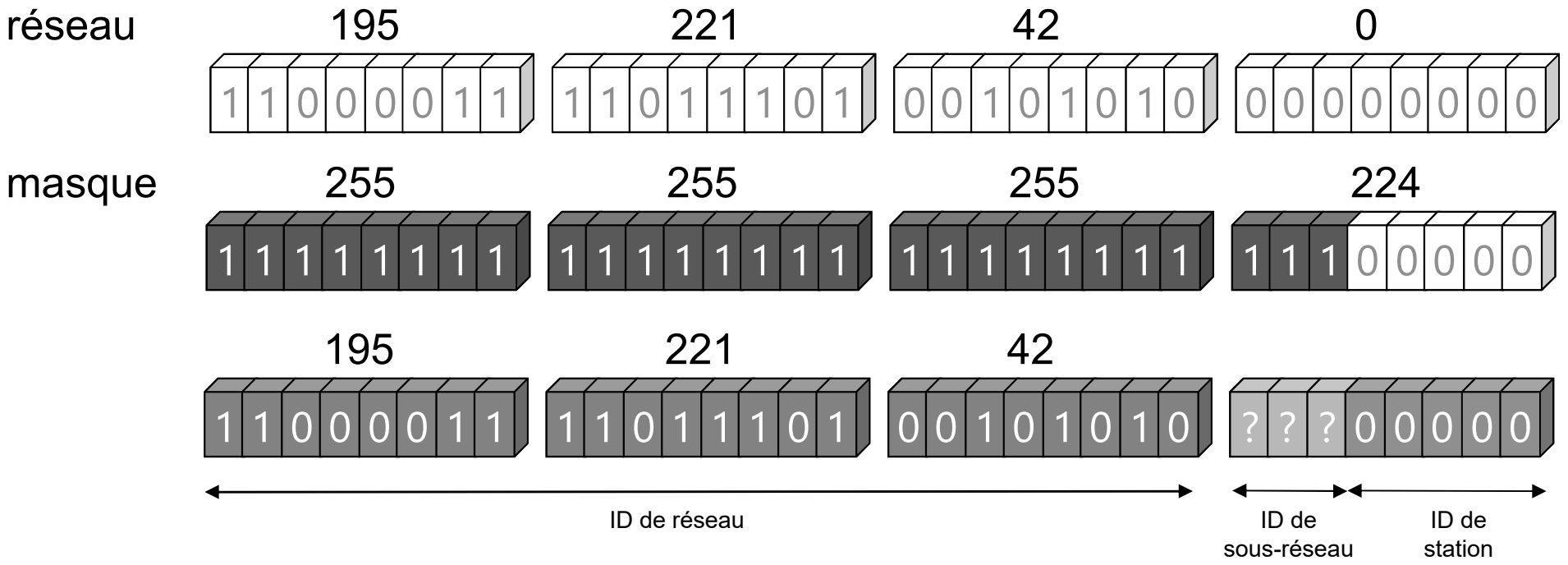


  - masque de réseau par défaut 255.255.0.0 si aucun sous-réseau n'est défini
  - masque 255.255.255.0 si présence de (au plus 254) sous-réseaux (de 254 hôtes chacun)

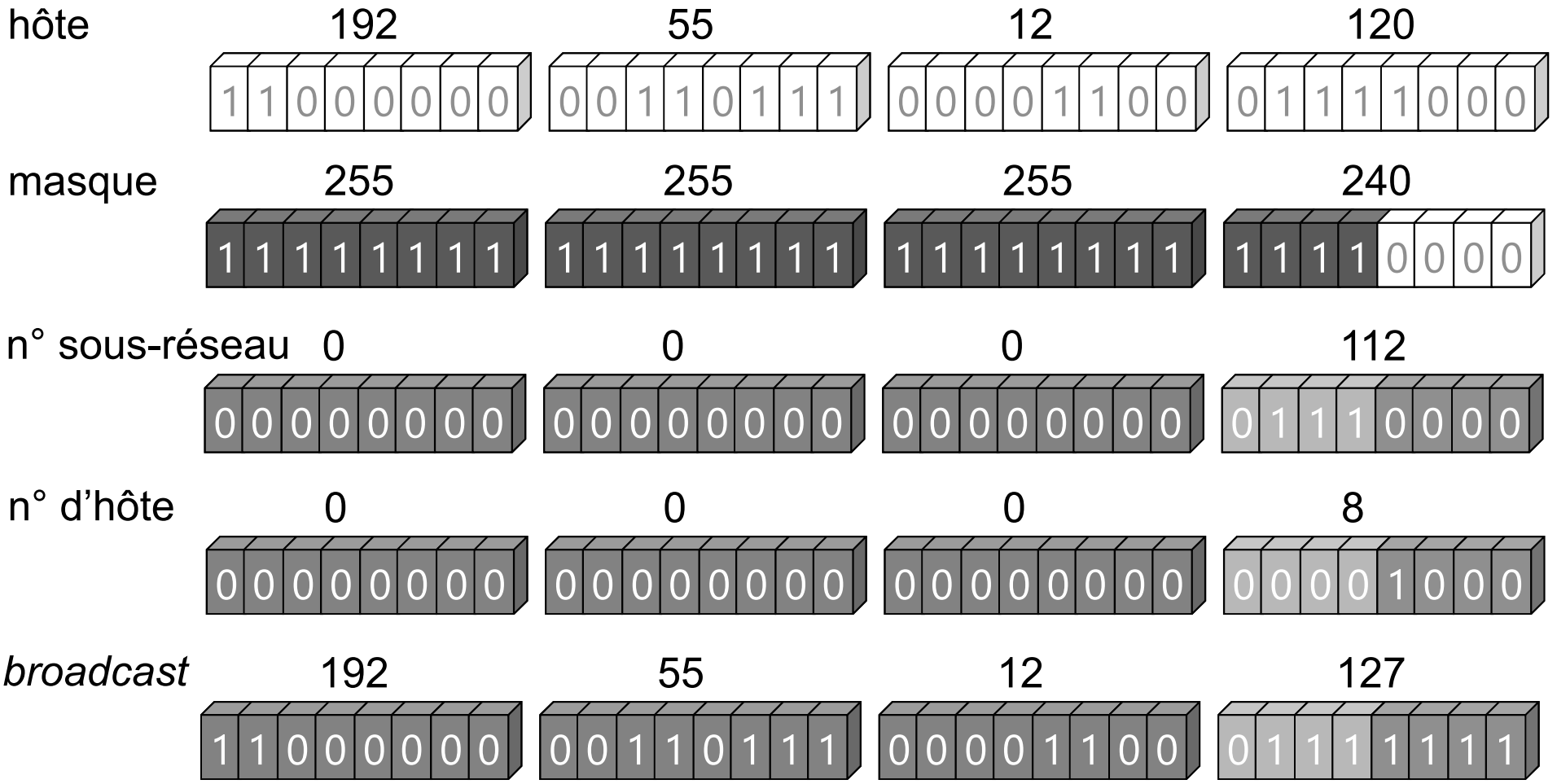
Exemple sans sous-réseaux



Exemple avec sous-réseaux

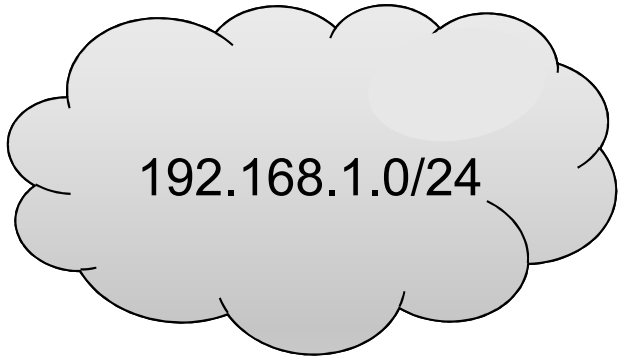


Exemple 2

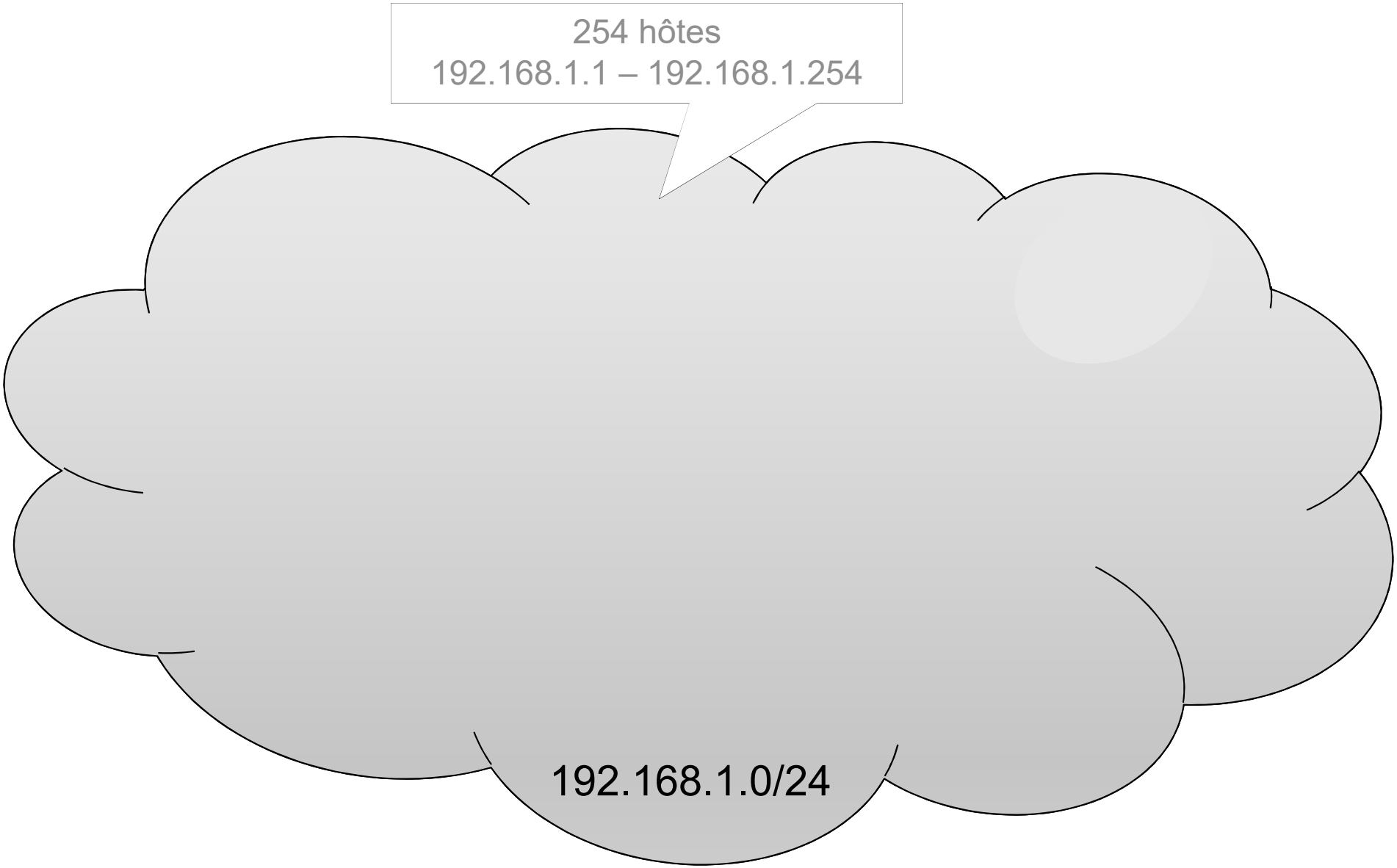

hôte
192
55
12
120
1 1 0 0 0 0 0 0
0 0 1 1 0 1 1 1
0 0 0 0 1 1 0 0
0 1 1 1 1 0 0 0
masque
255
255
255
240
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 0
n° sous-réseau
0
0
0
112
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
n° d'hôte
0
0
0
8
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0
broadcast
192
55
12
127
1 1 0 0 0 0 0 0
0 0 1 1 0 1 1 1
0 0 0 0 1 1 0 0
0 1 1 1 1 1 1 1

Découpage en sous-réseaux

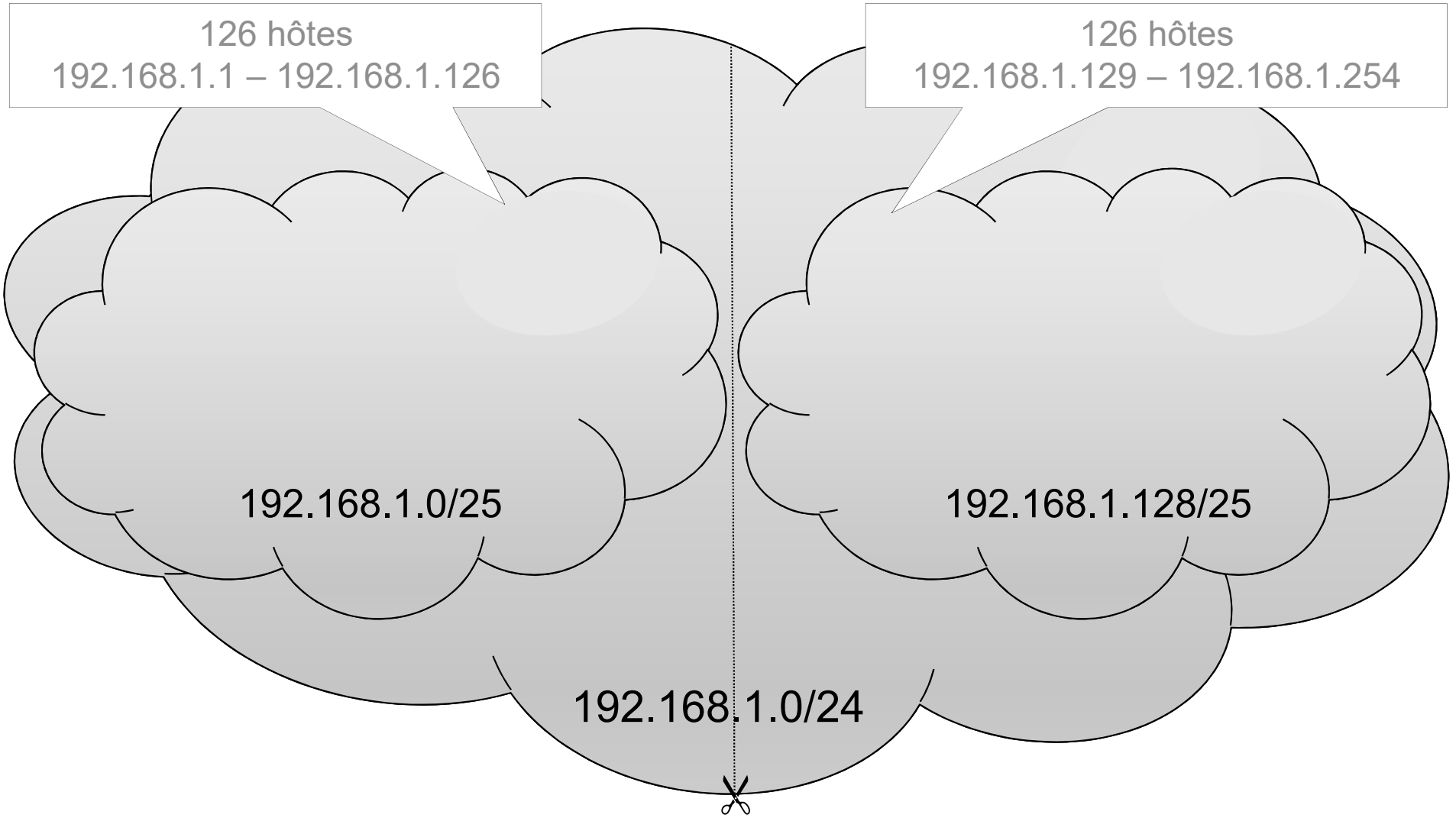
- Comment découper le réseau 192.168.1.0/24 en 3 sous-réseaux pour avoir :
  - 2 sous-réseaux d'au plus 60 machines
  - 1 sous-réseau d'au plus 124 machines
- Le nombre d'hôtes est différent pour au moins un sous-réseau
  - ➔ on ne peut pas utiliser le même masque



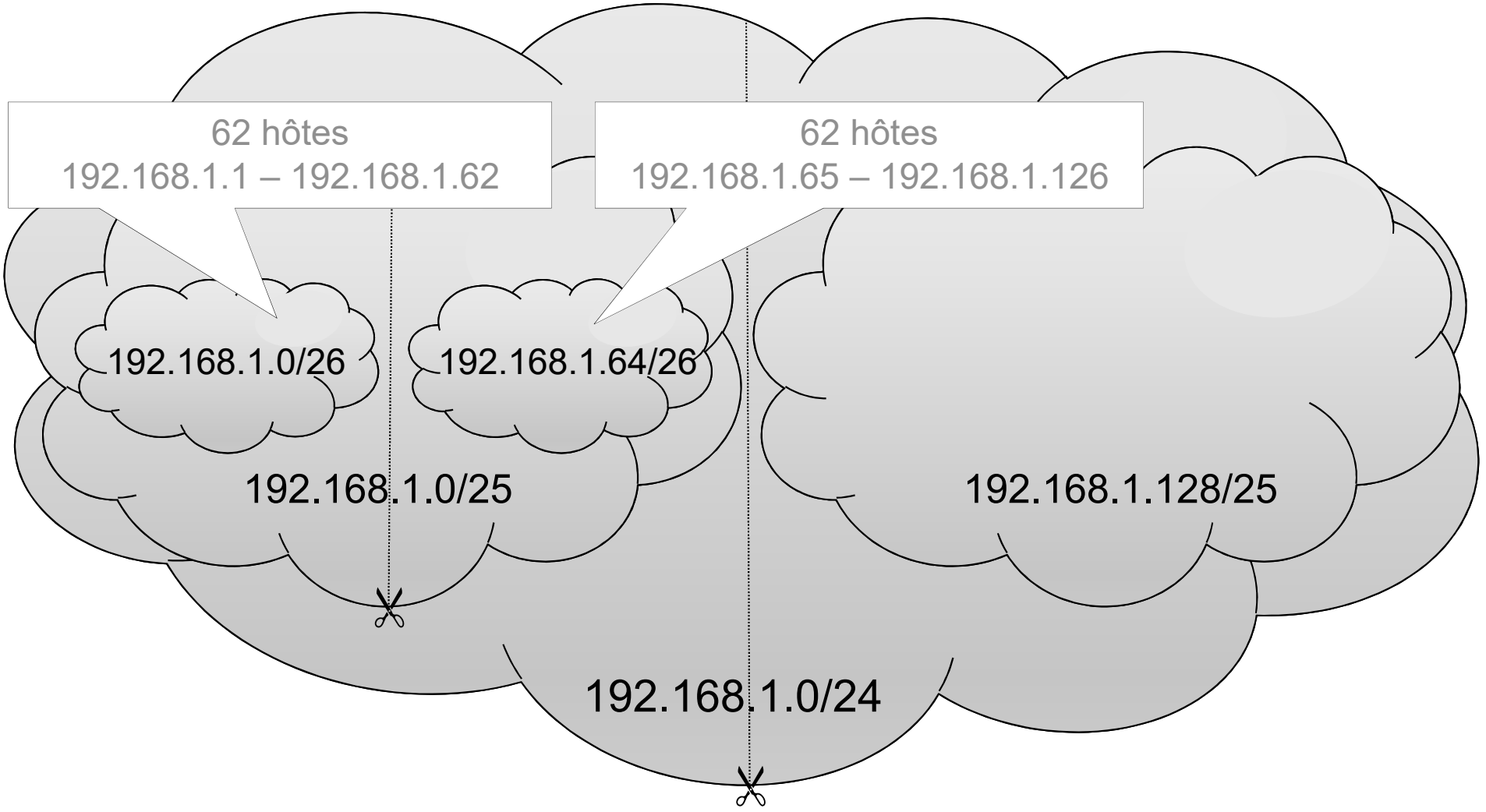
Découpage en sous-réseaux



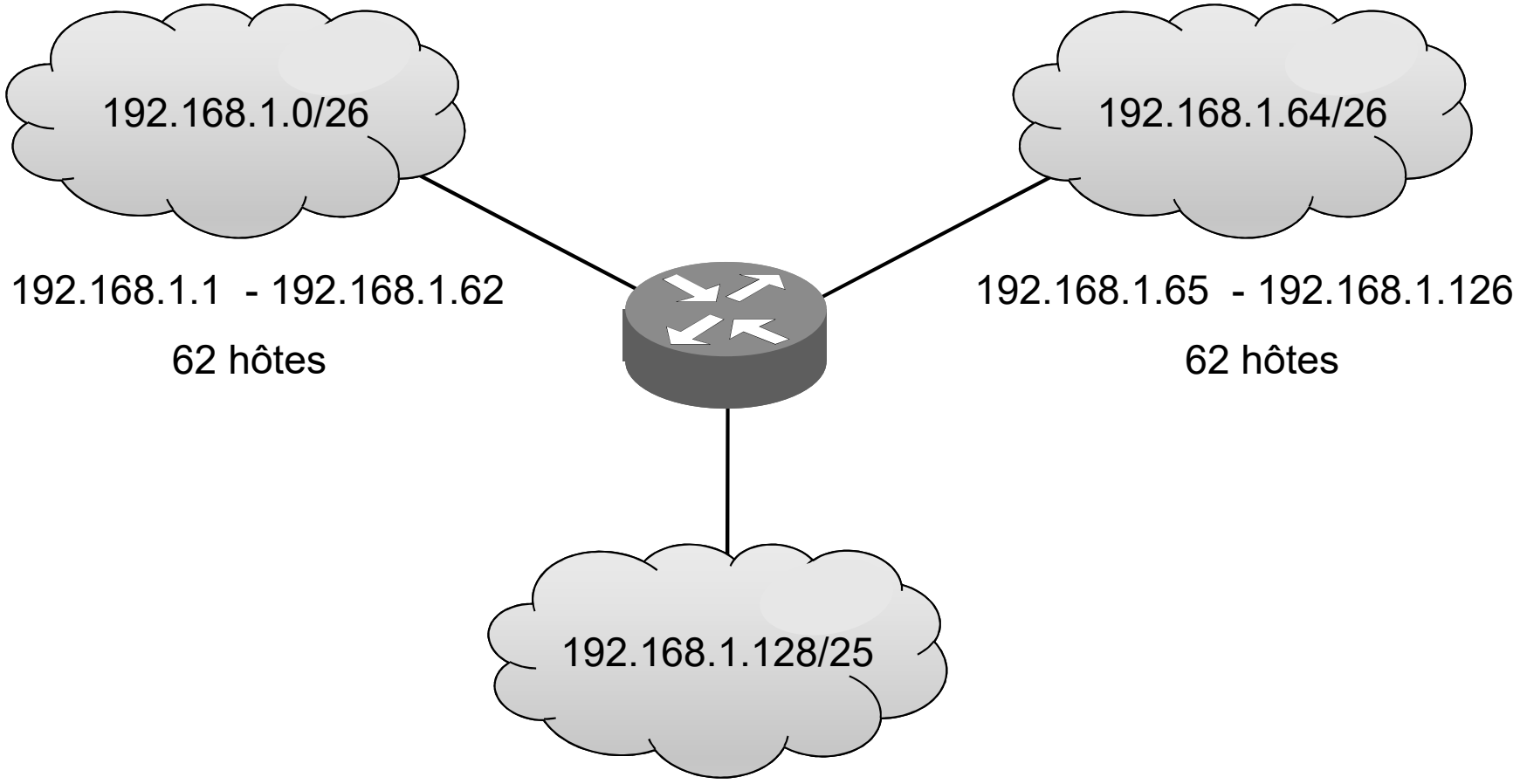
Découpage en sous-réseaux



Découpage en sous-réseaux



Découpage en sous-réseaux

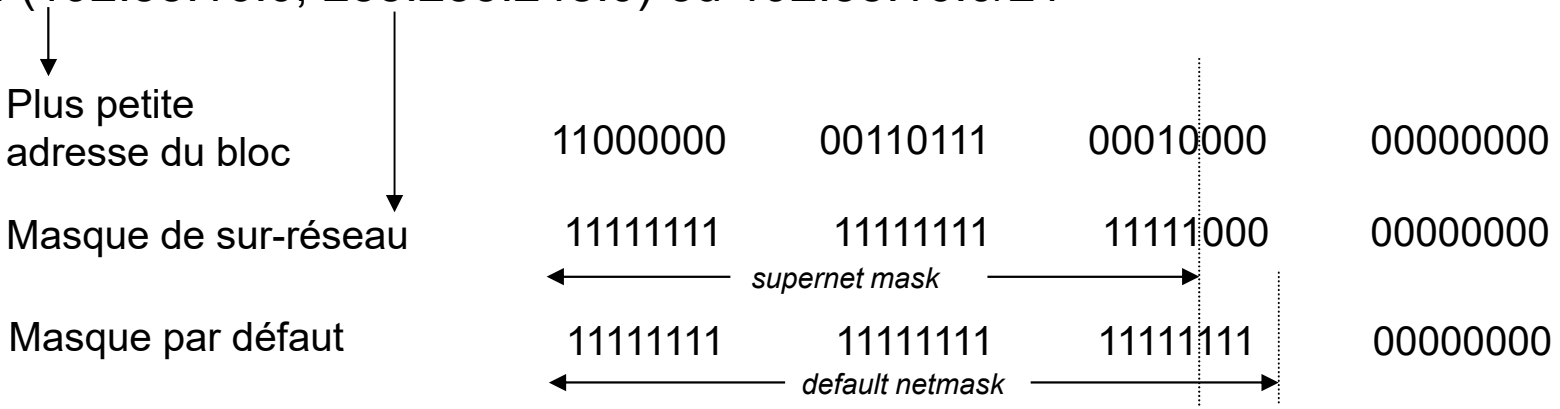


Sur-réseaux IP (v4)

- Regroupement de plusieurs réseaux en un seul bloc
  - concept développé en 1985 pour optimiser l'espace d'adressage IP
  - pénurie d'adresses de classe A et B, mais encore beaucoup d'adresses de classe C
  - consiste à affecter un bloc d'adresses de classes C plutôt qu'une adresse de classe B unique
  - principalement conçu pour les fournisseurs d'accès Internet (FAI)

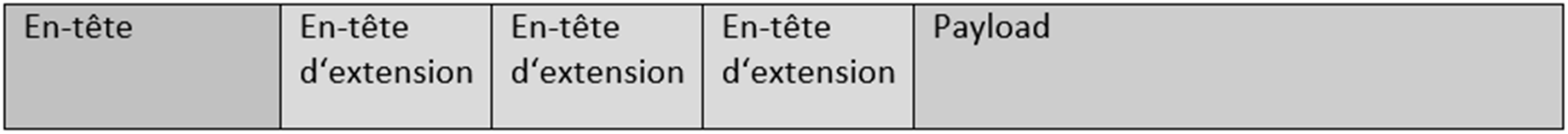
Adressage de sur-réseaux

- Problème de la taille de la table de routage (1 bloc de 256 adresses de classe C = 1 adresse de classe B $\Rightarrow$ 256 adresses de réseau)
- Technique du CIDR (*Classless Internet Domain Routing*) ou routage de domaine Internet sans classe
- Résume un bloc d'adresses de classe C en une seule entrée de table de routage
- Notation :
  - (plus basse adresse du bloc, masque de sur-réseau) ou
  - plus basse adresse du bloc / nombre de bits de préfixe commun
- exemple : (192.55.16.0, 255.255.248.0) ou 192.55.16.0/21

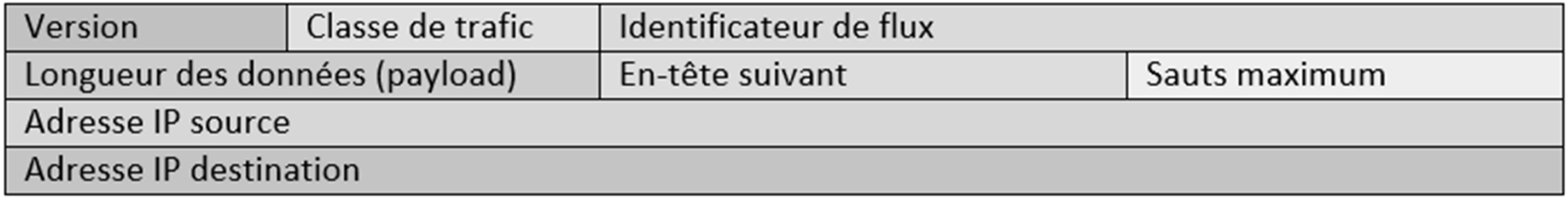


Format du paquet IPv6

- IPv6 utilise un format de paquet simplifié
  - En-tête de 40 bytes (320 bits) pour simplifier le traitement des paquets
  - Les informations optionnelles se retrouvent dans les **en-têtes d'extension**, insérées entre l'en-tête et le payload



  - Permet d'insérer des options sans avoir à modifier l'en-tête
- L'en-tête de paquet IPv6 ne comporte que 8 champs en-têtes



Adresses IPv6

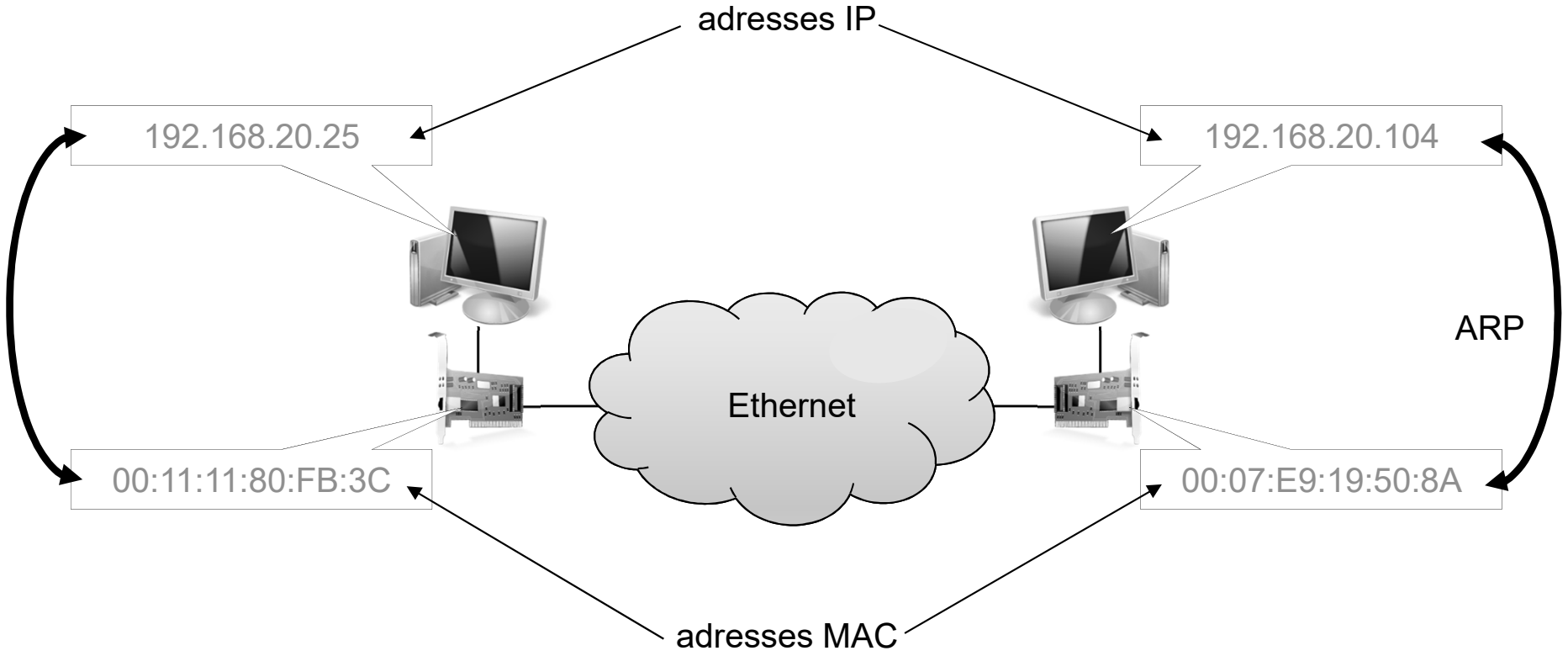
- Adresses codées sur 16 octets (128 bits contre 32)
  - offrant un espace d'adressage quasi illimité (environ $3{,}4 \times 10^{38}$ adresses)
  - notées sous forme de 8 groupes de 4 chiffres hexadécimaux séparés avec le symbole deux-points

    8000:0000:0000:0000:0123:4567:89AB:CDEF

Les protocoles ARP et ICMP

ARP (*Address Resolution Protocol*)

- Correspondance entre une adresse Internet (32 bits) et une adresse « physique » (Ethernet sur 48 bits)



ARP (*Address Resolution Protocol*)

- Lors de l'envoi d'un datagramme IP
  - on connaît l'adresse IP destination
  - on ne connaît pas l'adresse Ethernet

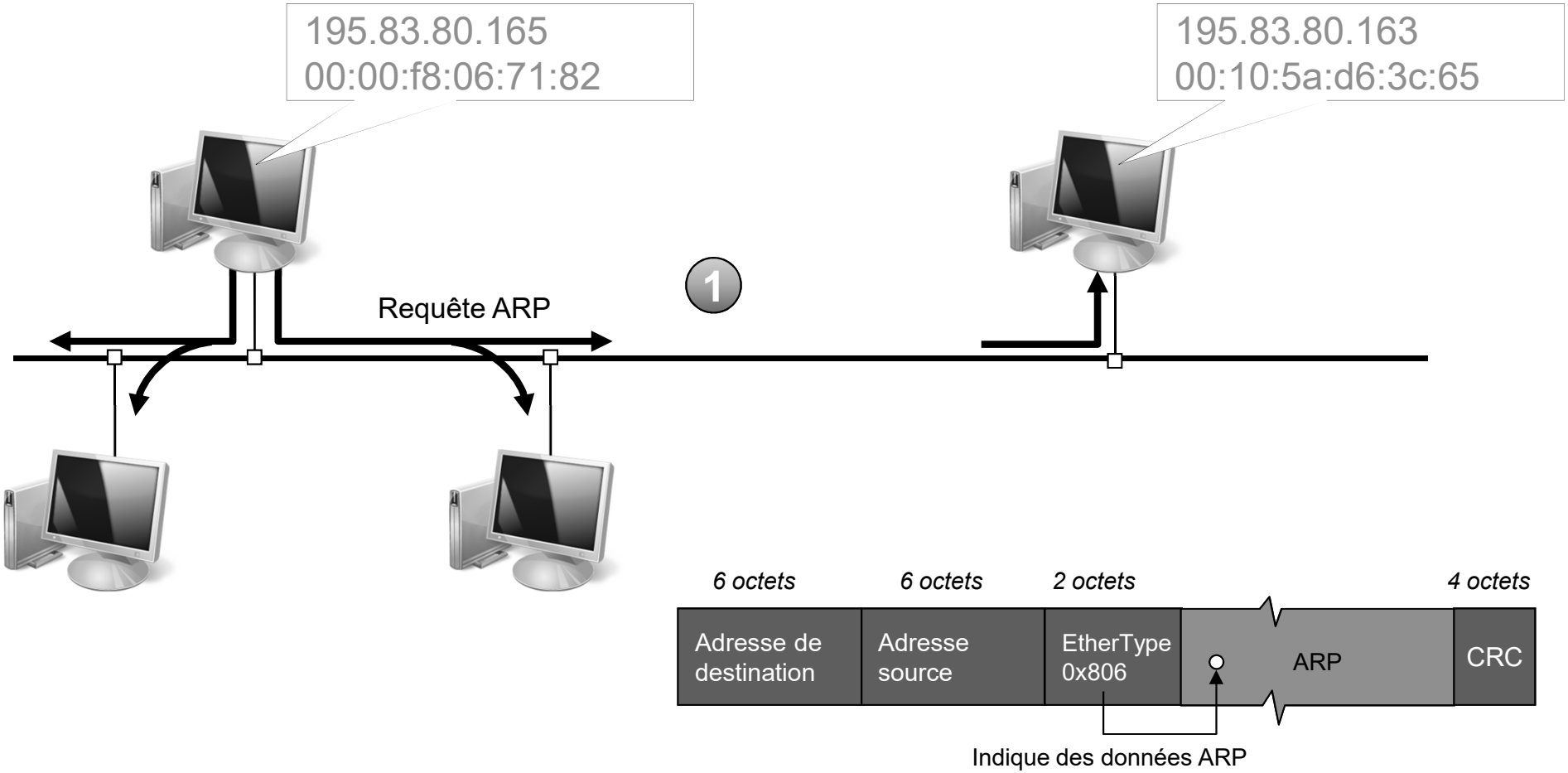
  ➔ protocole ARP
- Au boot d'une machine sans disque (Terminal X par exemple)
  - on connaît l'adresse Ethernet
  - on ne connaît pas l'adresse IP

  ➔ protocole RARP (*Reverse ARP*)

ARP (*Address Resolution Protocol*)

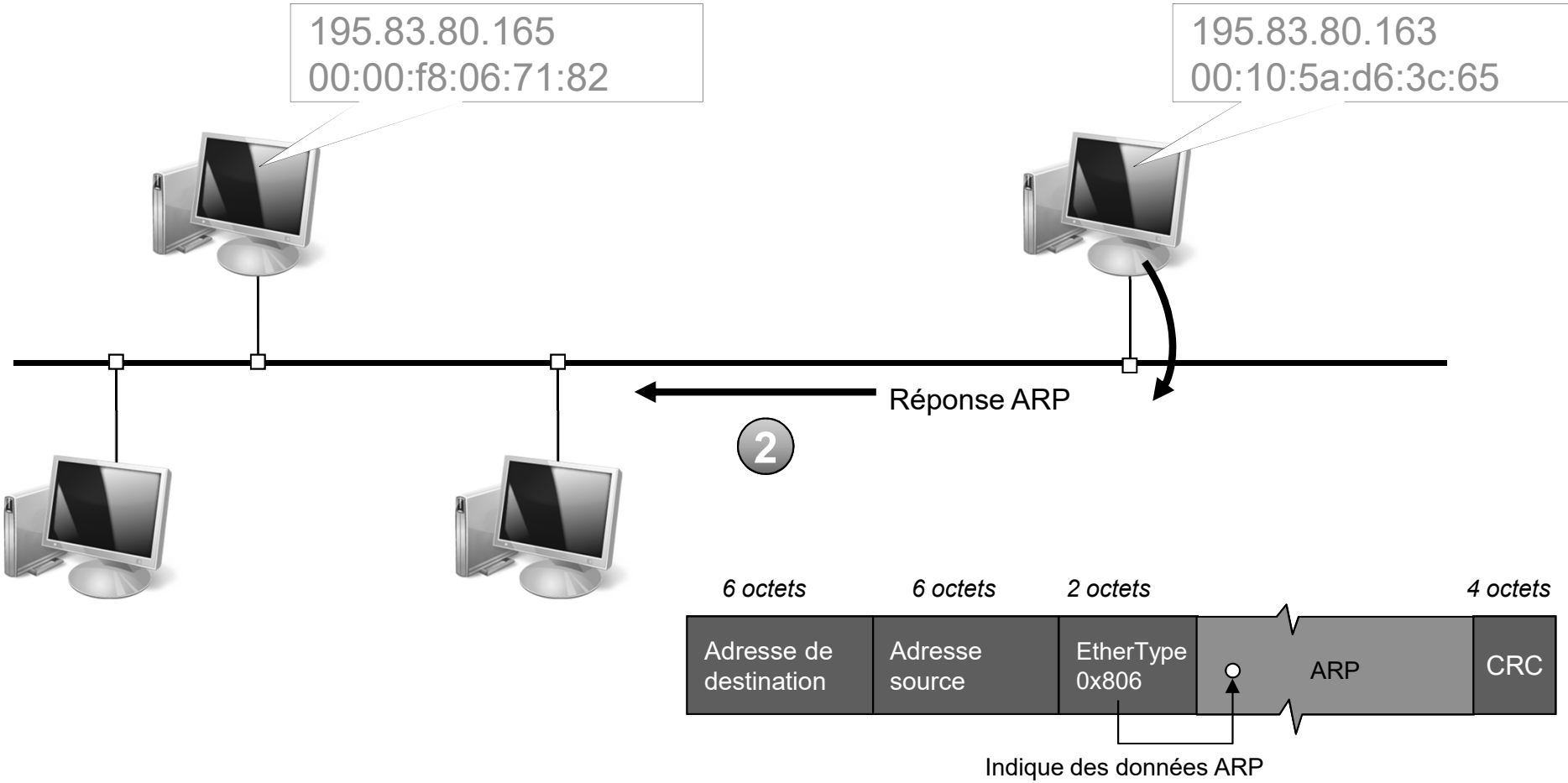
- A d'@ IP Ia, et d'@ Ethernet Ea veut envoyer un message à B d'@ IP Ib
  - A diffuse un message ARP avec l'adresse de diffusion matérielle (FF:FF:FF:FF:FF:FF)
  - Toutes les machines reçoivent la requête
  - Seul B renvoie un message contenant son adresse physique Eb
  - A met à jour sa table ARP en mémorisant Ib $\leftrightarrow$ Eb
  - B met à jour sa table ARP en mémorisant Ia $\leftrightarrow$ Ea

ARP (*Address Resolution Protocol*)




```
1 00:00:f8:06:71:82 ff:ff:ff:ff:ff:ff  ARP  Who has 195.83.80.163? Tell 195.83.80.165
```

ARP (*Address Resolution Protocol*)




```
1 00:00:f8:06:71:82 ff:ff:ff:ff:ff:ff  ARP  Who has 195.83.80.163? Tell 195.83.80.165
2 00:10:5a:d6:3c:65 00:00:f8:06:71:82  ARP  195.83.80.163 is at 00:10:5a:d6:3c:65
```

Protocole ICMP

- *Internet Control Message Protocol*
- Implémenté sur tous les équipements
- Message peut être envoyé par la destination ou n'importe quel équipement entre la source et la destination
  - en cas de problème dans un datagramme
  - pour demander à l'émetteur qu'il change son comportement
- Jamais de réponse à un message ICMP pour ne pas engendrer d'autres messages en cascade
- Messages ICMP encapsulés dans des datagrammes IP

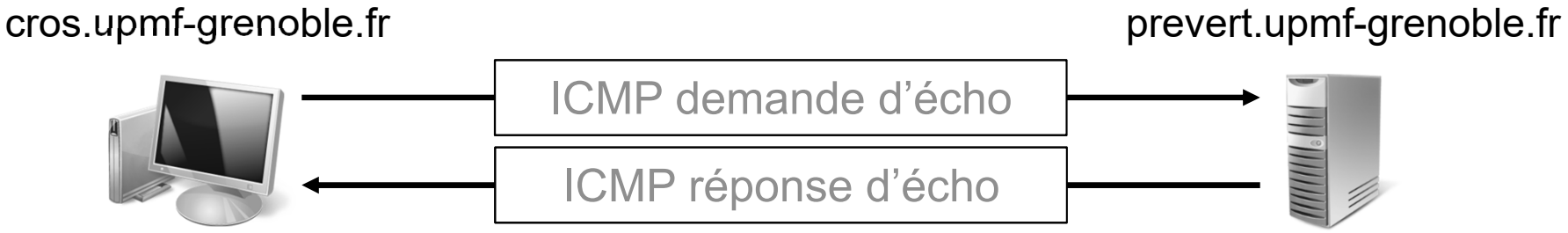
Protocole ICMP

- *Internet Control Message Protocol*
- 15 messages utilisés
  - 10 informations
    - Demande d'écho / réponse d'écho
    - Messages de routeurs
    - Horodatage
  - 5 erreurs
    - Destination inaccessible
    - Temps dépassé
    - Divers
    - Redirection

Protocole ICMP

- La commande `ping`
  - envoie un message ICMP de demande d'écho
  - la destination renvoie un message ICMP de réponse d'écho
  - permet de savoir si une machine est en route et accessible
  - mesure le temps moyen aller-retour à cette machine (RTT)

Protocole ICMP : ping

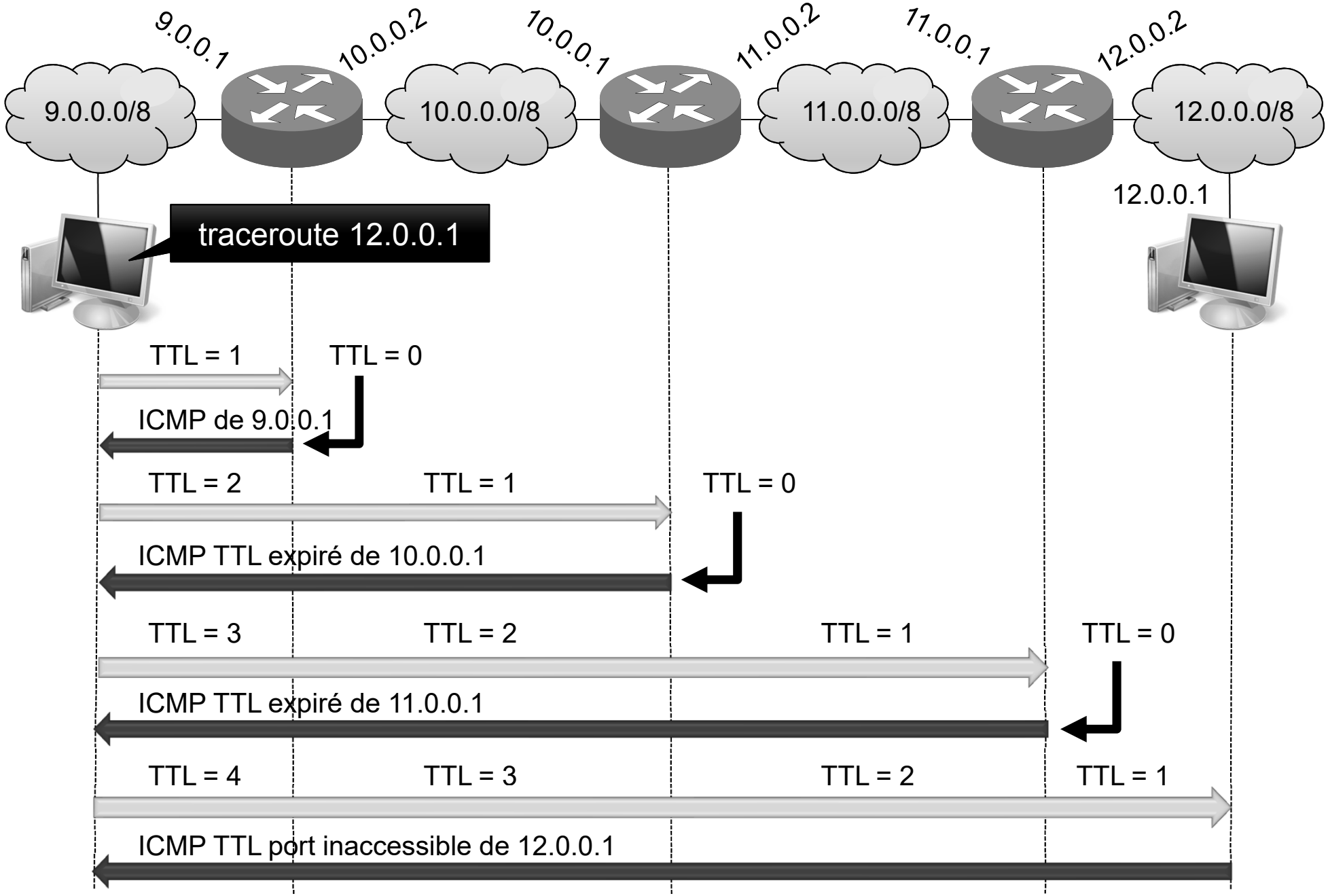



```
ping prevert.upmf-grenoble.fr

Envoi d'une requête 'ping' sur prevert.upmf-grenoble.fr [195.221.42.159]
avec 32 octets de données :
Réponse de 195.221.42.159 : octets=32 temps<10 ms TTL=128
Réponse de 195.221.42.159 : octets=32 temps<10 ms TTL=128
Réponse de 195.221.42.159 : octets=32 temps<10 ms TTL=128
Réponse de 195.221.42.159 : octets=32 temps<10 ms TTL=128

Statistiques ping pour 195.221.42.159:
     Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en milli-secondes :
     minimum = 2ms, maximum =  2ms, moyenne =  2ms
```

Protocole ICMP

- La commande `traceroute`
  - envoie un paquet UDP avec un TTL égal à 1
  - puis recommence en augmentant le TTL de 1 à chaque envoi
  - à chaque fois que le TTL arrive à 0, le routeur renvoie un message ICMP d'erreur (« *Time-to-live exceeded* »)
  - permet de connaître la route exacte empruntée

Protocole ICMP : traceroute



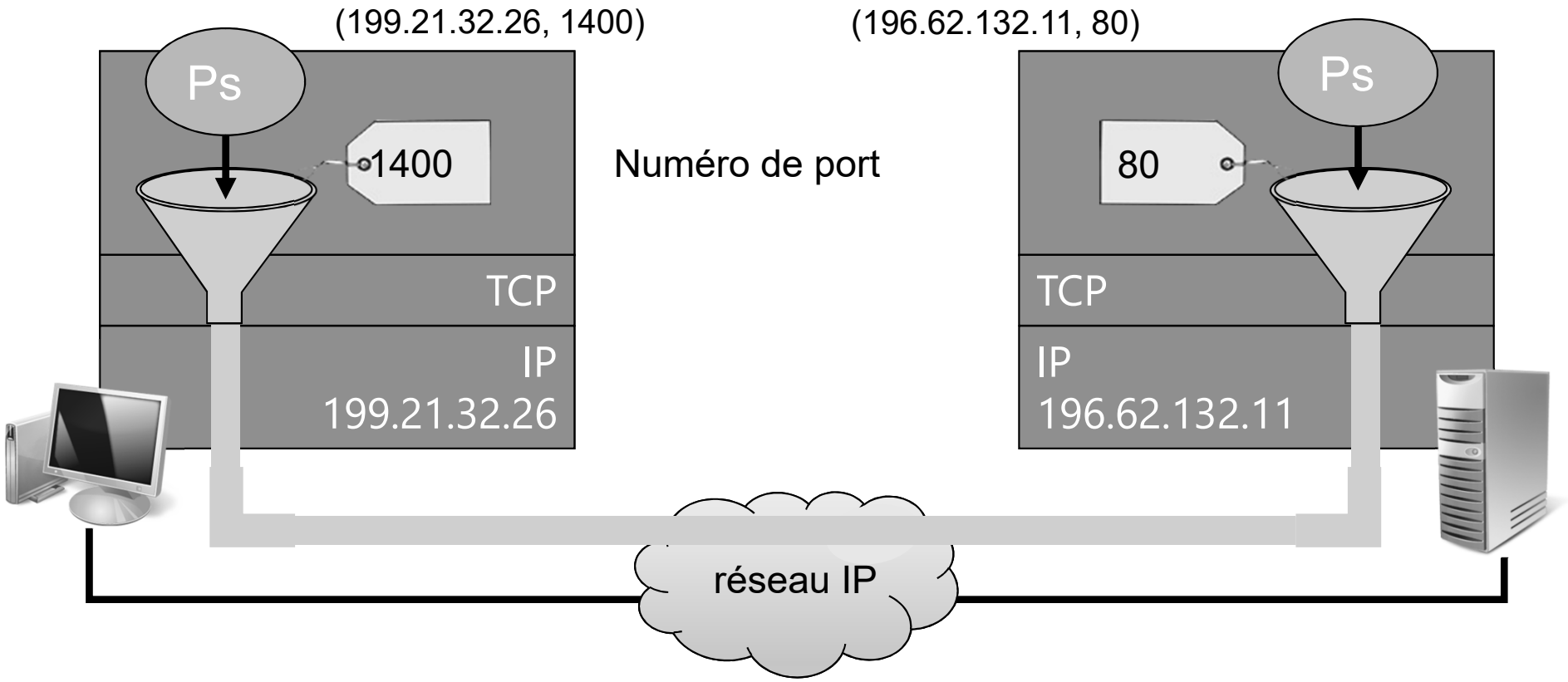
Les protocoles UDP et TCP

Couche Transport

- Deux protocoles pour la communication entre applications :
  - UDP : *User Datagram Protocol* (RFC 768)
    - mode sans connexion
    - pas de contrôle d'erreur (sans garantie)
  - TCP : *Transmission Control Protocol* (RFC 793)
    - protocole orienté connexion
    - offre de la fiabilité (pas de perte, pas d'erreur)
    - ordonné
    - contrôle de flux

Couche Transport

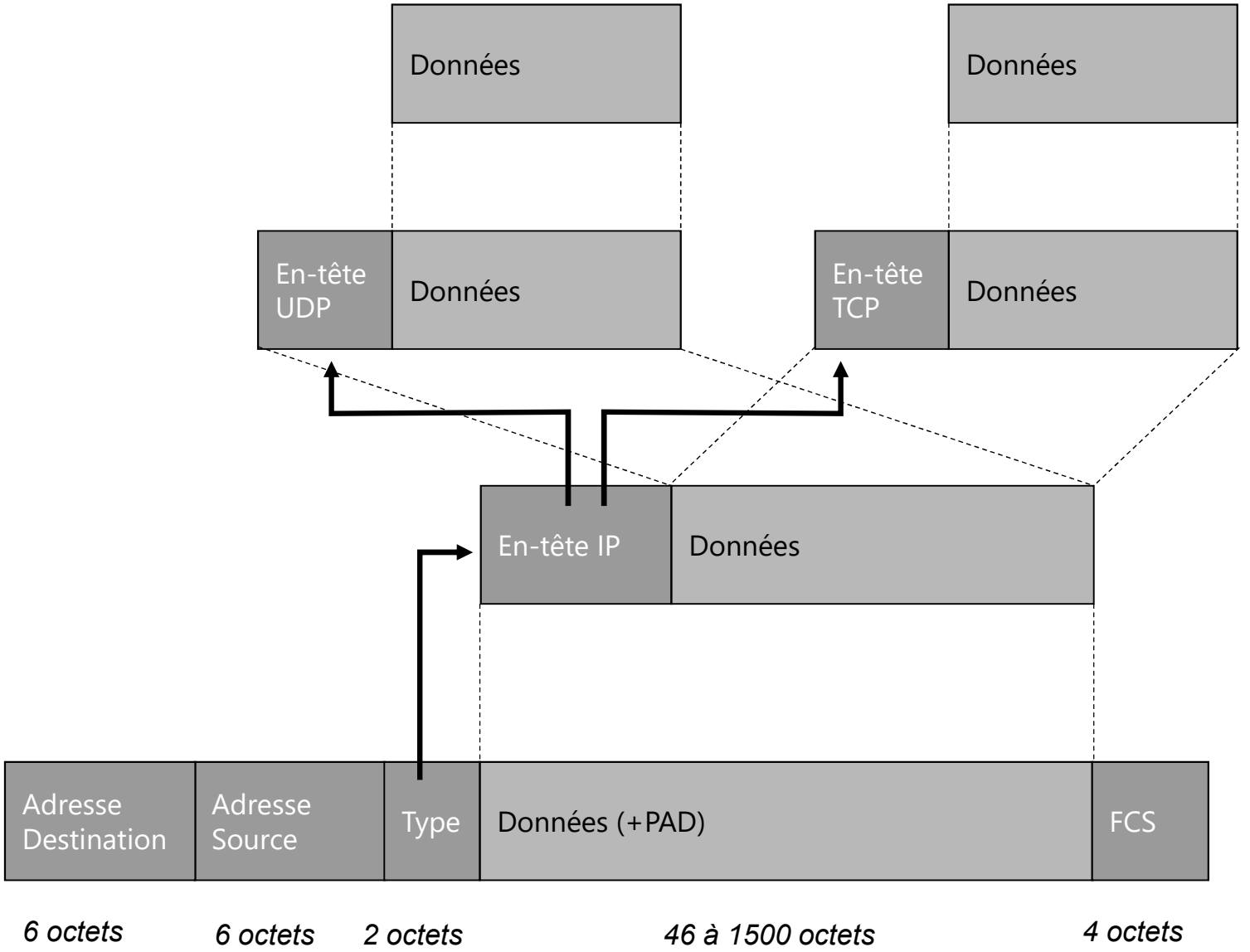
- Identification d'une application par un numéro de port
- **socket** : combinaison d'une adresse IP et d'un n° de port
- La combinaison de 2 sockets définit complètement une connexion TCP ou un échange UDP



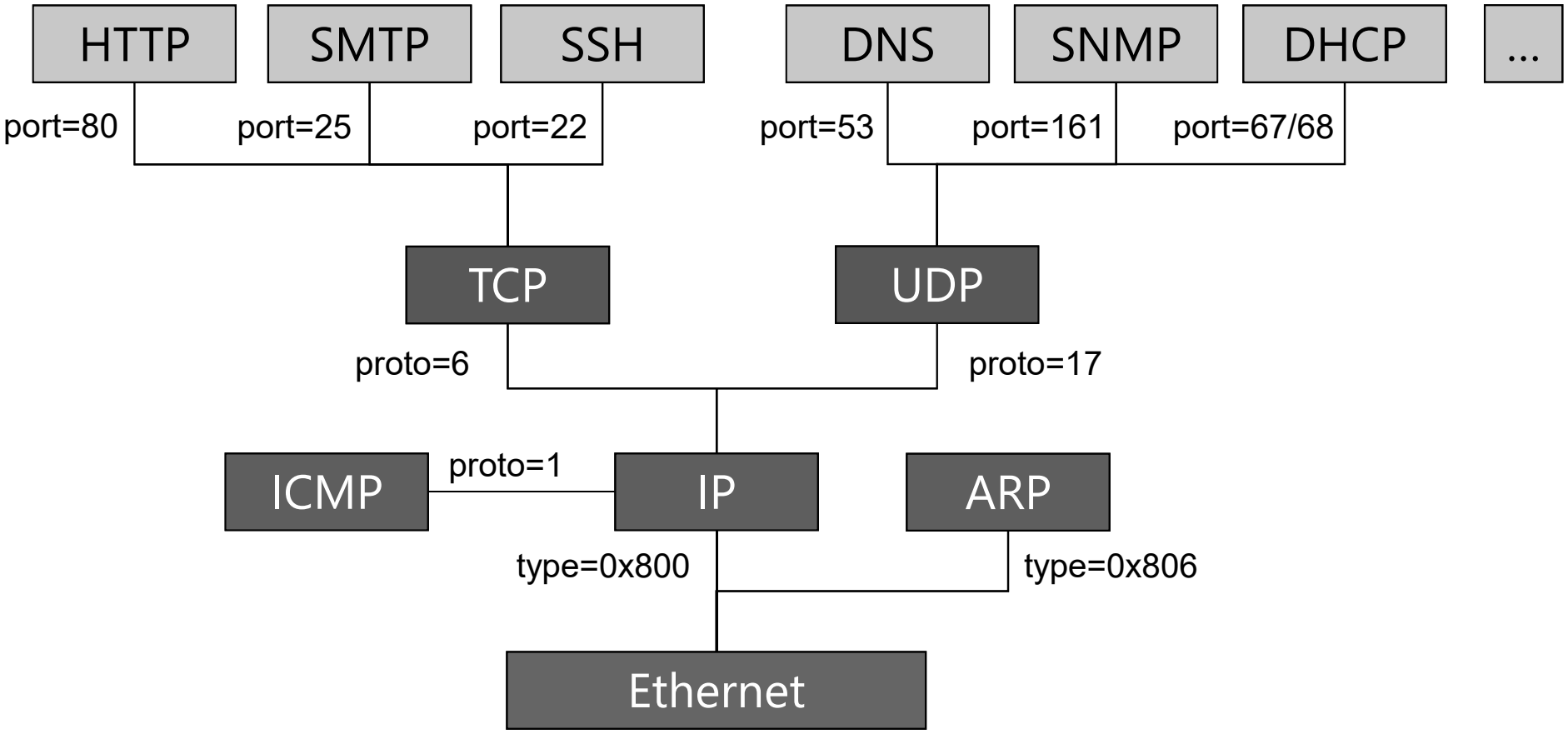
Notion de port

- Un service rendu par un programme serveur sur une machine est accessible par un port
- Un port est identifié par un entier (16 bits)
  - de 0 à 1023
    - ports **reconnus** ou **réservés**
    - sont assignés par l'IANA (*Internet Assigned Numbers Authority*)
    - donnent accès aux services standard : courrier (SMTP port 25), serveur web (HTTP port 80) …
  - > 1024
    - ports « **utilisateurs** » disponibles pour placer un service applicatif quelconque
- Un service est souvent connu par un nom (FTP, ...)
  - La correspondance entre nom et numéro de port est donnée par le fichier **`/etc/services`**
    - **`80 ⇔ http`**
    - **`25 ⇔ smtp`**
    - …

Encapsulation : rappel

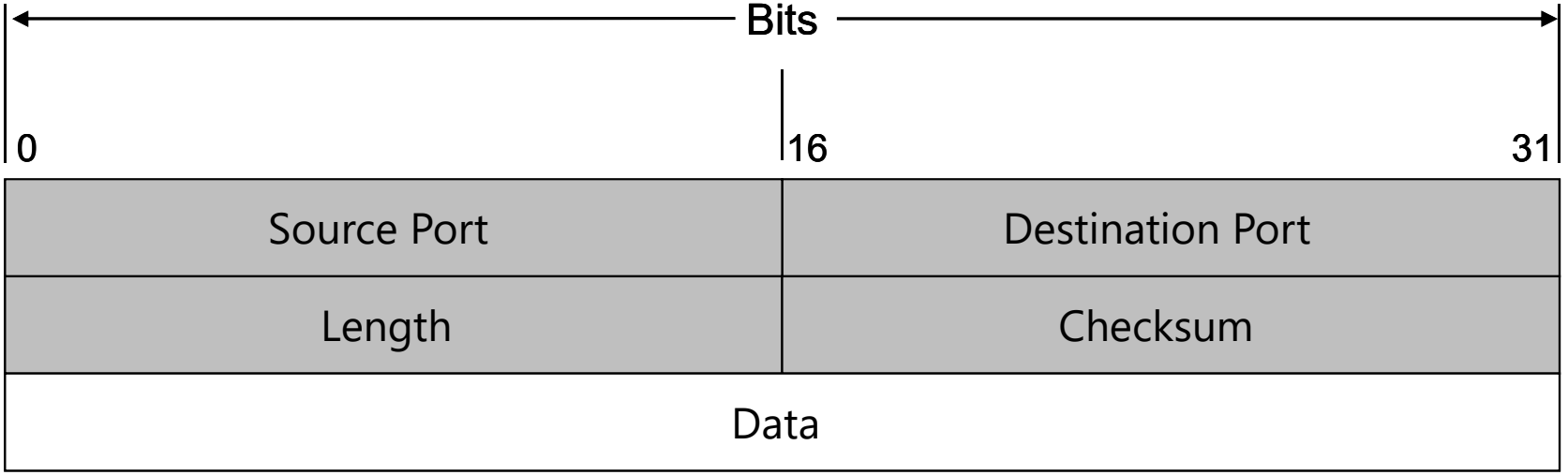


Identification des protocoles



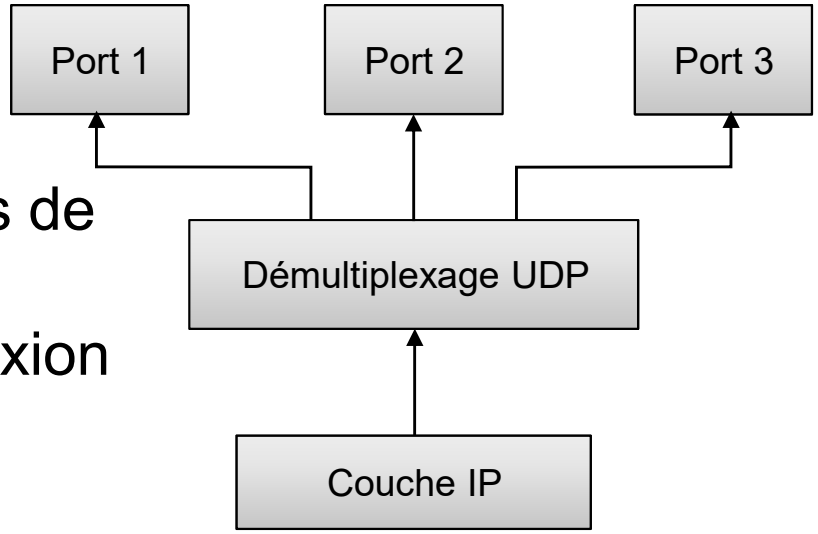
Format du paquet UDP

- User Datagram Protocol (RFC 768)
  - service sans connexion, sans garantie, utilisant IP pour le transport de messages



UDP

- Caractéristiques d'UDP
  - démultiplexage des données d'application à l'aide des numéros de ports
  - orienté datagrammes sans connexion
  - efficace pour les applications en diffusions/multidiffusion
  - plus rapide, plus simple que TCP mais moins robuste
- Ce qu'UDP ne fait pas :
  - retransmission en cas d'erreur,
  - séquencement des données
  - contrôle de flux



Format du paquet UDP

- Champ *Source Port*
  - indique numéro de port du processus émetteur ou celui où on peut adresser les réponses
  - 0 = aucun numéro de port attribué
- Champ *Destination Port*
  - identifie le processus correspondant à l'adresse IP de destination où envoyer les données
  - le datagramme UDP est rejeté et un message ICMP de type « port inaccessible » est envoyé si le port ne peut pas être contacté

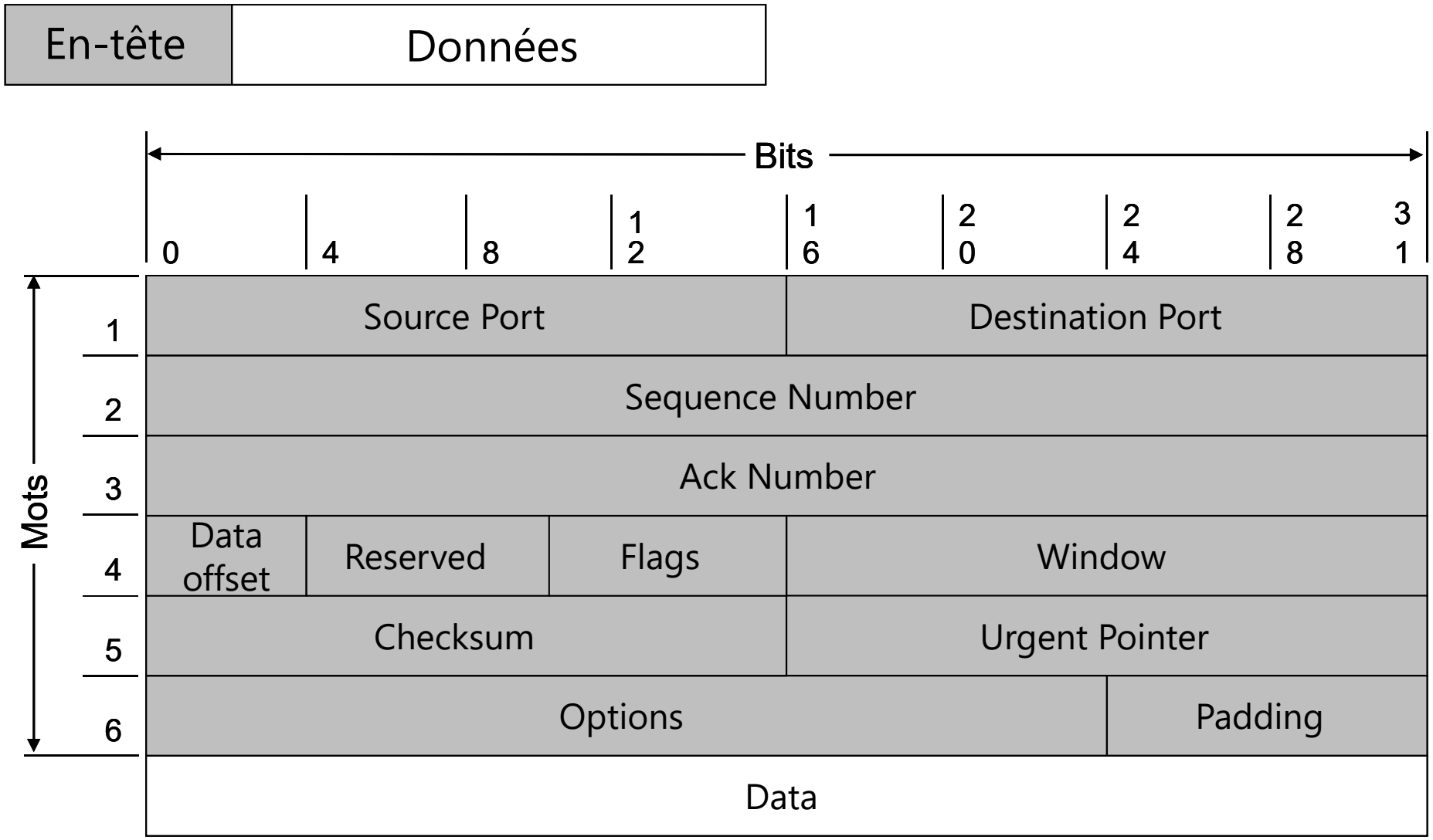
Format du paquet UDP

- Champ *Length*
  - contient la longueur du paquet UDP en octets (en-tête + données)
  - une valeur minimale de 8 correspond à un paquet de données vides
- Champ *Checksum*
  - calculé sur un pseudo en-tête constitué
    - des adresses IP source et destination, du code de protocole (17) et de la longueur du paquet UDP
    - l'en-tête UDP elle-même
    - des données

TCP

- Le protocole TCP n'est exécuté que par la machine source et la machine destination (pas dans les routeurs)
- Caractéristiques :
  - transport de bout en bout
  - mode connecté : ouverture, fermeture, gestion de connexion
  - sans erreur : contrôle et retransmission si nécessaire
  - sans perte : « numérotation » et retransmission
  - ordonnée : préservation du séquencement
  - système d'acquittement
  - contrôle de flux (fenêtre d'émission) en full-duplex
  - identification du service par numéro de port

Format du segment TCP



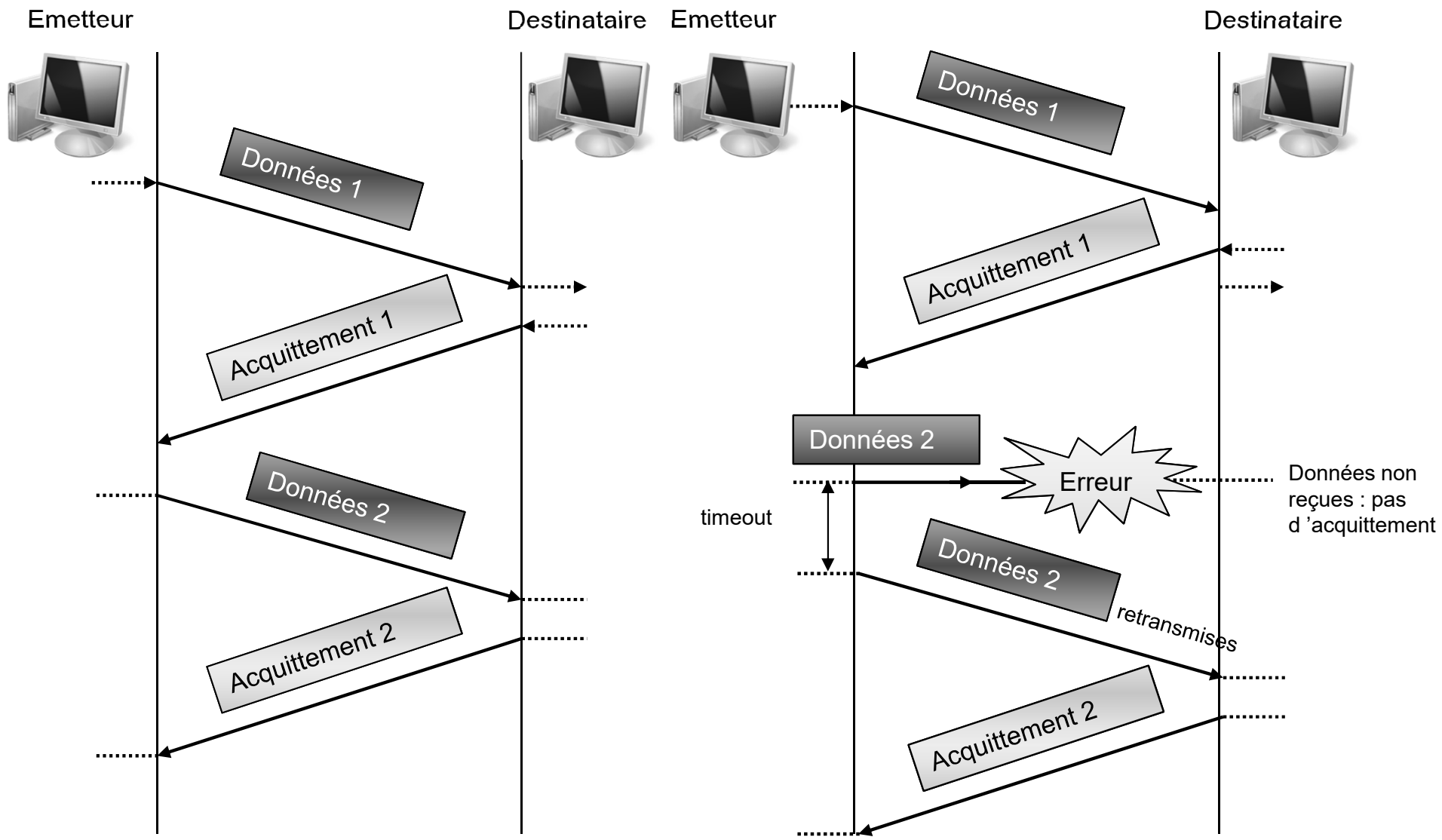
Format du segment TCP

- Champ *Source Port* et *Destination Port*
  - indiquent le numéro de port du processus émetteur et celui auquel sont destinées les données
- Champ *Data Offset*
  - Indique la taille de l'en-tête TCP en nombre de mots de 32 bits

Fiabilité de la transmission

- Assurée par un mécanisme appelé PAR (*Positive Acknowledgment with Retransmission*)
  - Un numéro de séquence (*Sequence Number*) est attribué à chaque octet de données
  - Un accusé de réception **positif** (*Ack Number*) est envoyé à l'émetteur lorsque les octets ont été correctement reçus
    - La vérification est faite à l'aide du total de contrôle
    - Les segments TCP endommagés sont éliminés par le destinataire
  - Au bout d'un délai d'attente déterminé, le module TCP d'envoi retransmet les segments pour lesquels aucun accusé de réception positif n'a été reçu

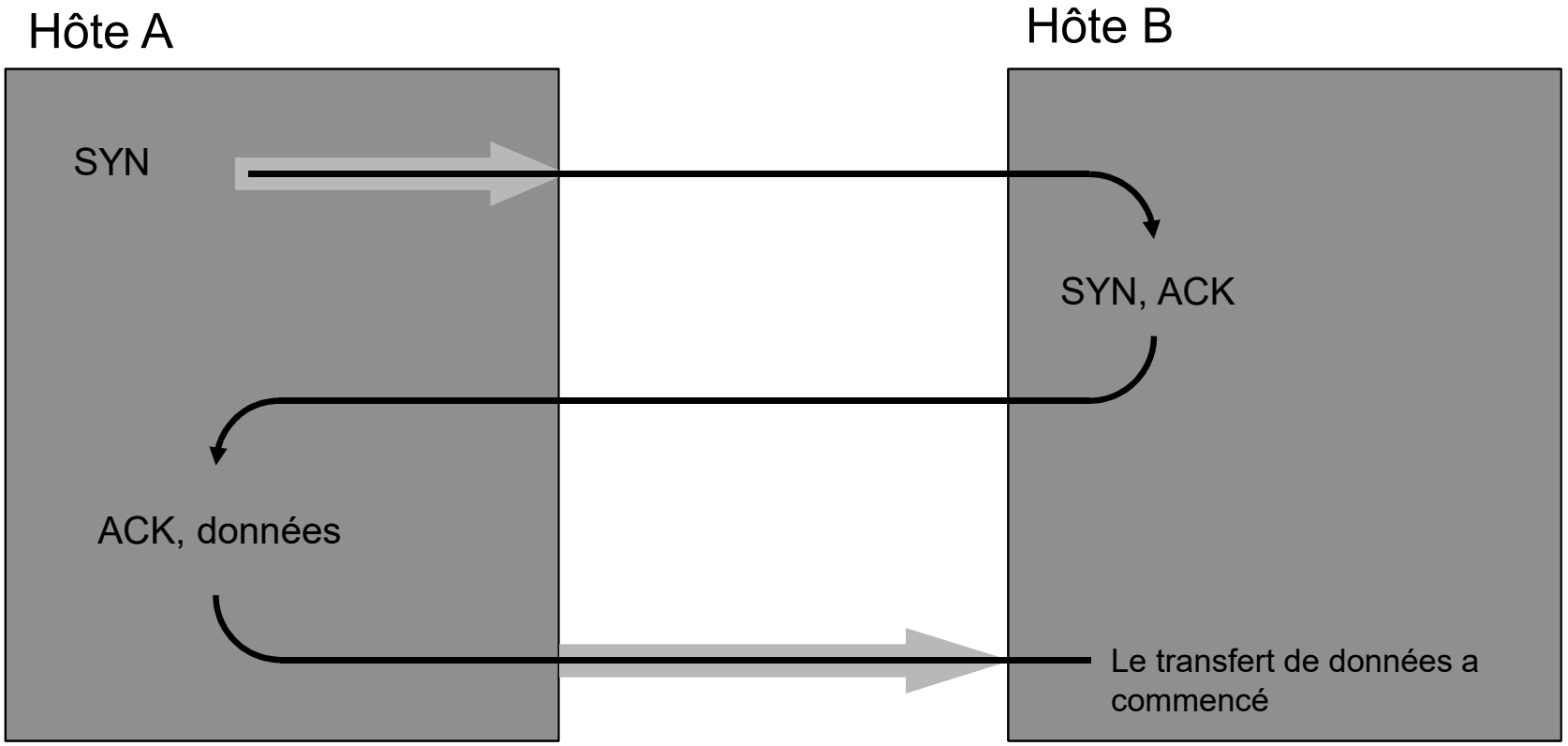
Mécanisme PAR



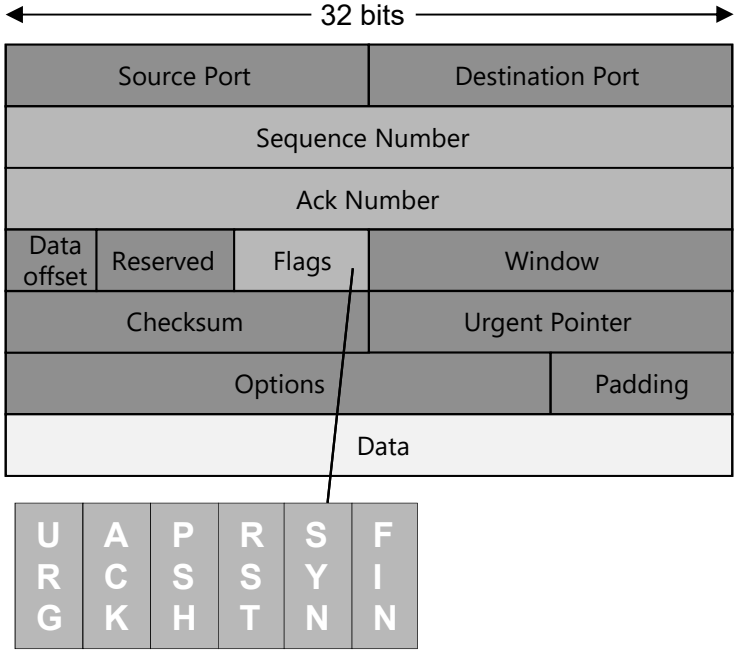
Mécanisme de la poignée de main

- TCP établit une **connexion logique** de bout en bout entre deux machines-hôtes communicantes
  - Avant tout transfert de données, une information de contrôle appelée *poignée de main*, est échangée entre les deux extrémités
  - Ce mécanisme met en œuvre 3 échanges de segment (*three-way handshake*) dans lesquels les bits **SYN** et **ACK** du champ *Flags* sont positionnés
  - Une fois la connexion établie, les données sont transmises
  - Les deux extrémités s'échangent une autre poignée de main pour indiquer la fin de la connexion (segments avec le bit **FIN** et **ACK** positionnés)

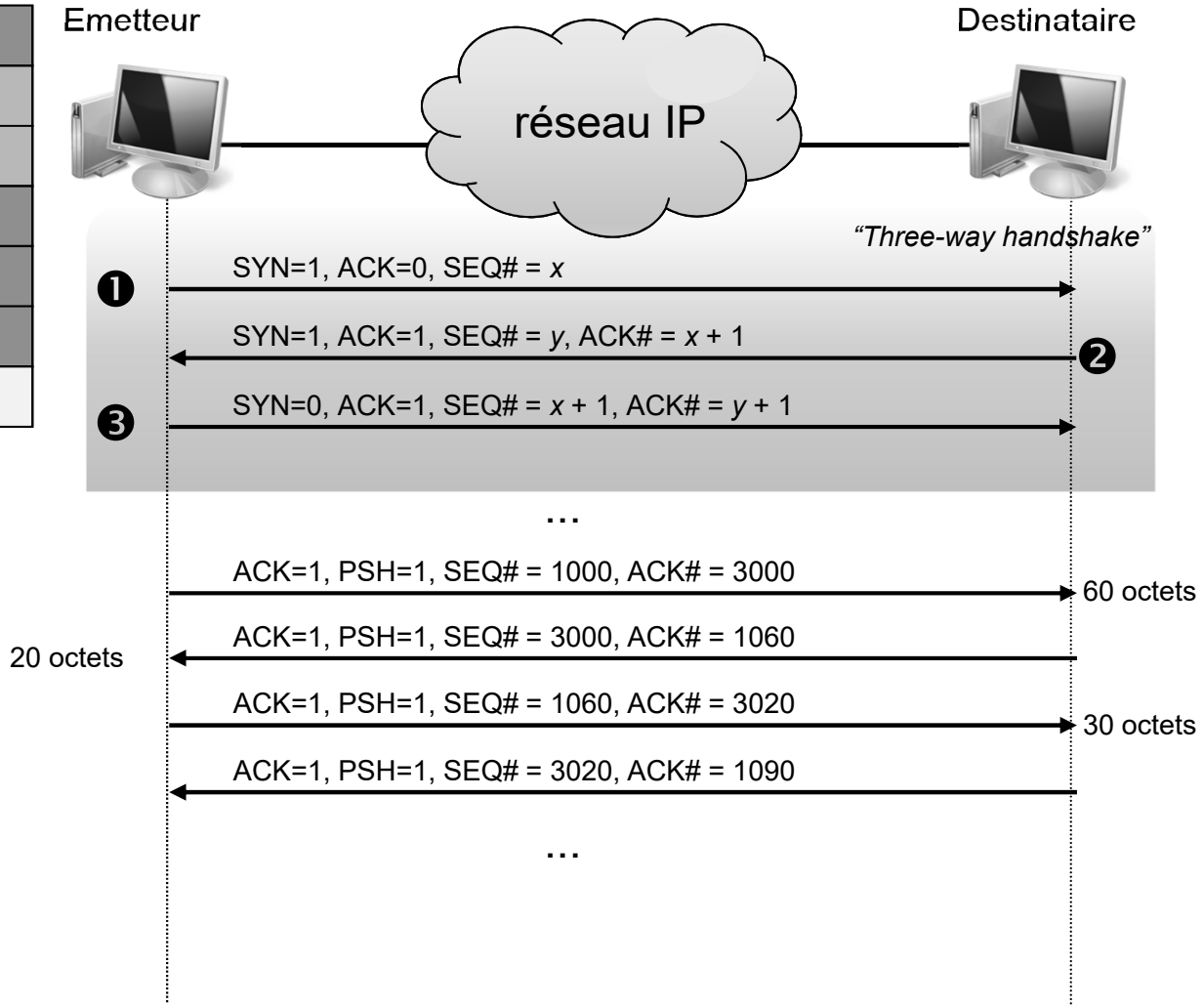
Mécanisme de la poignée de main



Transfert des données

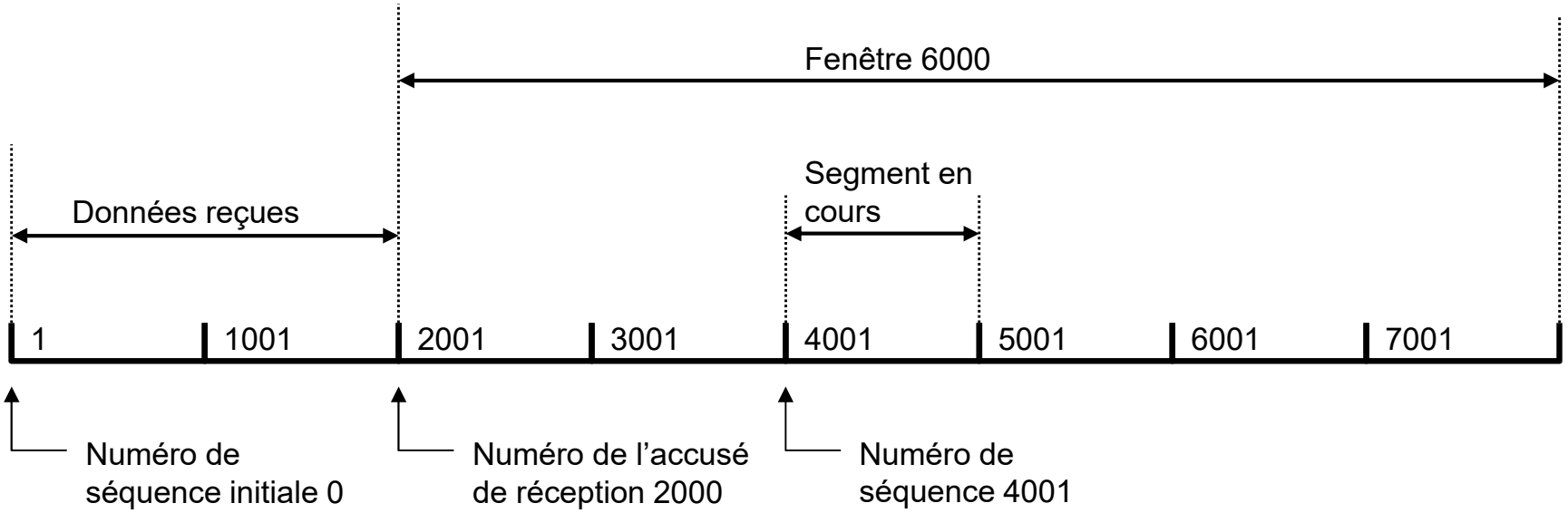


- **URG** : tenir compte du champ *Urgent Pointer*
- **ACK** : tenir compte du champ *Acknowledgment Number*
- **PSH** : délivrer immédiatement les données au processus de couche supérieure
- **RST** (reset) : réinitialiser la connexion
- **SYN** : établir une connexion
- **FIN** : terminer la connexion

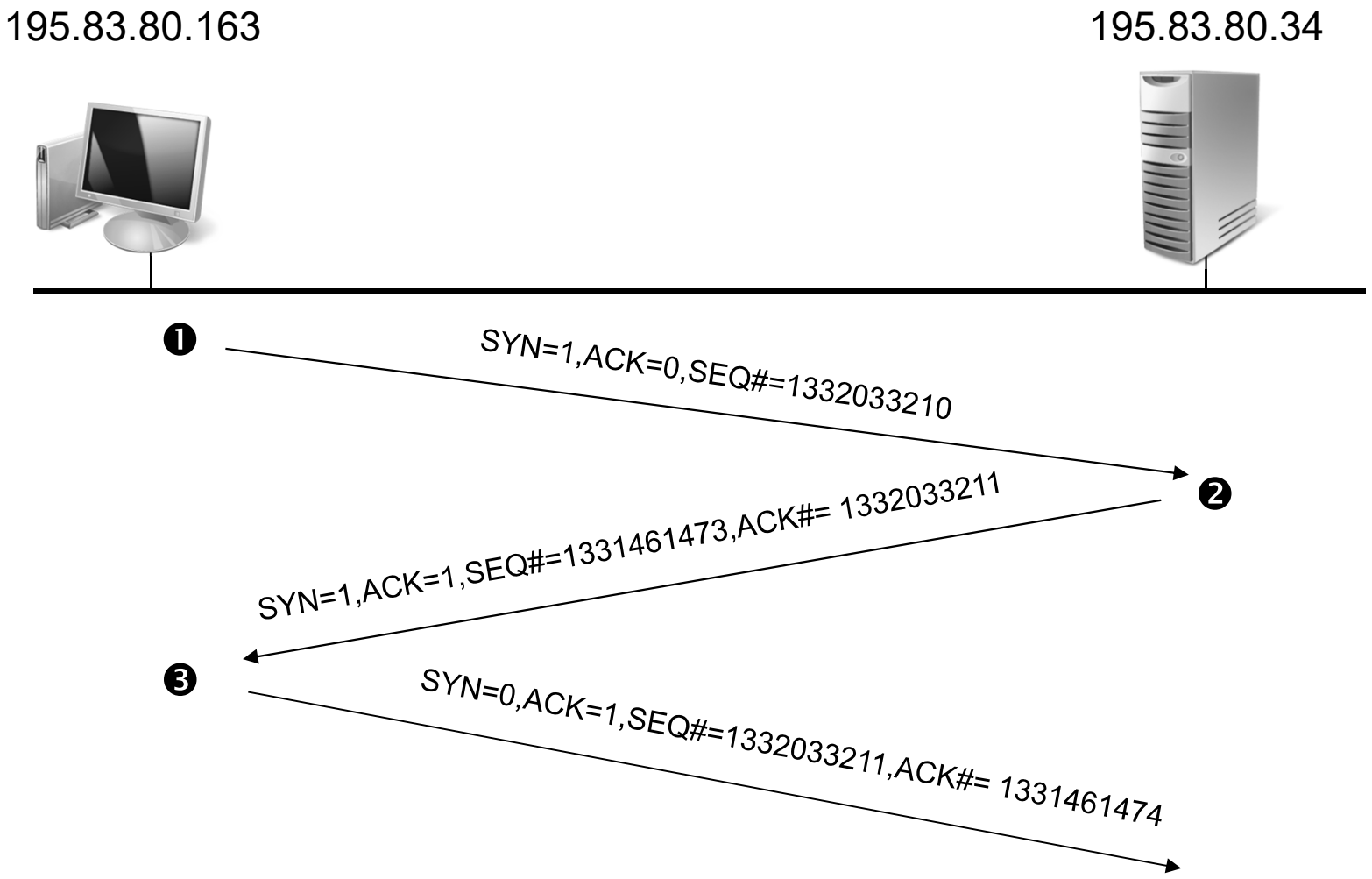


Contrôle de flux

- Le champ *Window* contient le nombre d'octets que l'extrémité distante peut recevoir
  - Le contrôle de flux est réalisé en modifiant la taille de la fenêtre
  - Une fenêtre de taille nulle indique à l'émetteur d'interrompre la transmission jusqu'à la réception d'une valeur non nulle



Trace de protocole TCP



Trame (1)

```
Frame Number: 1
Packet Length: 74 bytes
Ethernet II
Internet Protocol
Transmission Control Protocol
    Source port: 1026 (1026)
    Destination port: telnet (23)
    Sequence number: 1332033210
    Header length: 40 bytes
    Flags: 0x0002 (SYN)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...0 .... = Acknowledgment: Not set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x583a (correct)
    Options: (20 bytes)
        Maximum segment size: 1460 bytes
        SACK permitted
        Time stamp: tsval 75607, tsecr 0
        NOP
        Window scale: 0 bytes

 0  00e0 1692 1582 0010 5ad6 3c65 0800 4500   ........Z.<e..E.
10  003c 0000 4000 4006 1350 c353 50a3 c353   .<..@.@..P.SP..S
20  5022 0402 0017 4f65 36ba 0000 0000 a002   P"....Oe6.......
30  16d0 583a 0000 0204 05b4 0402 080a 0001   ..X:............
40  2757 0000 0000 0103 0300                  'W........
```

Trame (2)

```
Frame Number: 2
Packet Length: 74 bytes
Ethernet II
Internet Protocol
Transmission Control Protocol
    Source port: telnet (23)
    Destination port: 1026 (1026)
    Sequence number: 1331461473
    Acknowledgment number: 1332033211
    Header length: 40 bytes
    Flags: 0x0012 (SYN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..1. = Syn: Set
        .... ...0 = Fin: Not set
    Window size: 32120
    Checksum: 0x5b42 (correct)
    Options: (20 bytes)
        Maximum segment size: 1460 bytes
        SACK permitted
        Time stamp: tsval 587834999, tsecr 75607
        NOP
        Window scale: 0 bytes
 0  0010 5ad6 3c65 00e0 1692 1582 0800 4500   ..Z.<e........E.
10  003c a614 4000 3f06 6e3b c353 5022 c353   .<..@.?.n;.SP".S
20  50a3 0017 0402 4f5c 7d61 4f65 36bb a012   P.....O\}aOe6...
30  7d78 5b42 0000 0204 05b4 0402 080a 2309   }x[B..........#.
40  a677 0001 2757 0103 0300                  .w..'W....
```

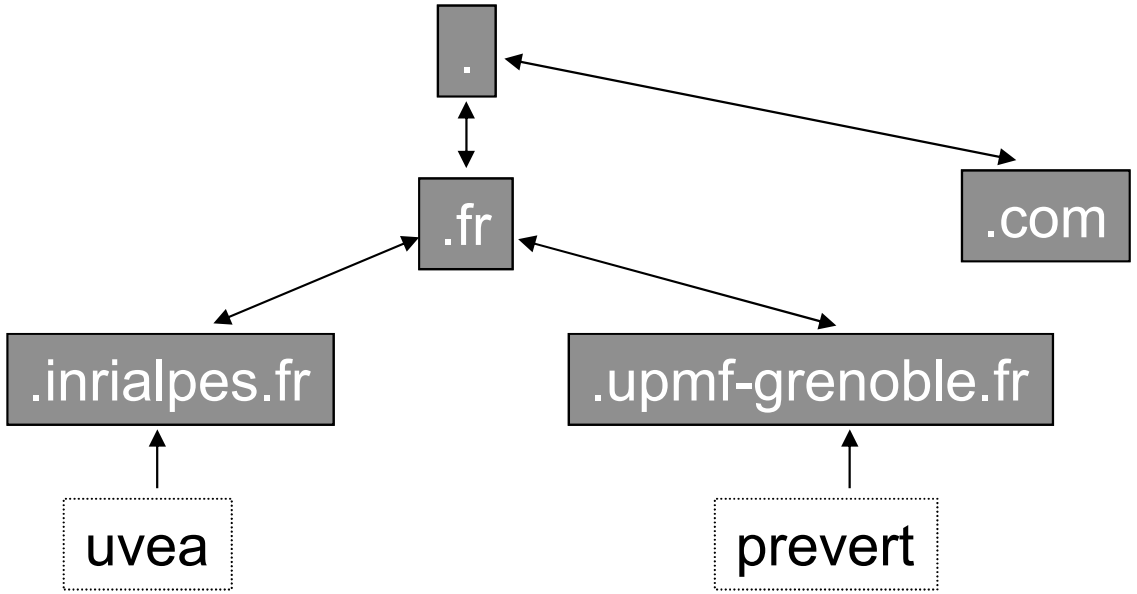
Trame (3)

```
Frame Number: 3
Packet Length: 74 bytes
Ethernet II
Internet Protocol
Transmission Control Protocol
    Source port: 1026 (1026)
    Destination port: telnet (23)
    Sequence number: 1332033211
    Acknowledgment number: 1331461474
    Header length: 32 bytes
    Flags: 0x0010 (SYN, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0xf0af (correct)
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 75607, tsecr 587834999

 0  00e0 1692 1582 0010 5ad6 3c65 0800 4500   ........Z.<e..E.
10  0034 0000 4000 4006 1358 c353 50a3 c353   .4..@.@..X.SP..S
20  5022 0402 0017 4f65 36bb 4f5c 7d62 8010   P"....Oe6.O\}b..
30  16d0 f0af 0000 0101 080a 0001 2757 2309   ............'W#.
40  a677                                      .w .w..'W....
```

Les services réseau

DNS (*Domain Name System*)

- Correspondance entre un nom et une adresse IP
- **Exemple :** prevert.umpf-grenoble.fr ⇔ 195.221.42.159
- Noms plus faciles à retenir que les adresses IP
- Hiérarchie de serveurs



FTP (*File Transfert Protocol*)

- Application pour transférer des fichiers (RFC 959)
- Utilise 2 connexions TCP (➔transfert fiable) :
  - 1 de contrôle (commandes et réponses) : port 21
  - 1 de transfert de données : port 20
    - cette connexion est ouverte puis fermée à chaque transfert
- Modes
  - client : processus d'un utilisateur, par exemple
  - serveur : démon ftpd traditionnellement lancé par inetd sous Unix
- Le client ouvre la connexion
- Le serveur attend

TELNET (*TErminaL NETwork protocol*)

- Terminal virtuel, remote terminal, terminal à distance (RFC854)
- Utilise une connexion TCP
  - Fiable mais gourmand en bande passante
  - Utilise le port 23 pour le serveur
- Modes
  - client : processus d'un utilisateur
  - serveur : démon telnetd qui est lancé par inetd sur Unix
- Le client ouvre la connexion

Le nom de login et le mot de passe sont transmis en clair (c'est-à-dire non chiffrés) sur le réseau !

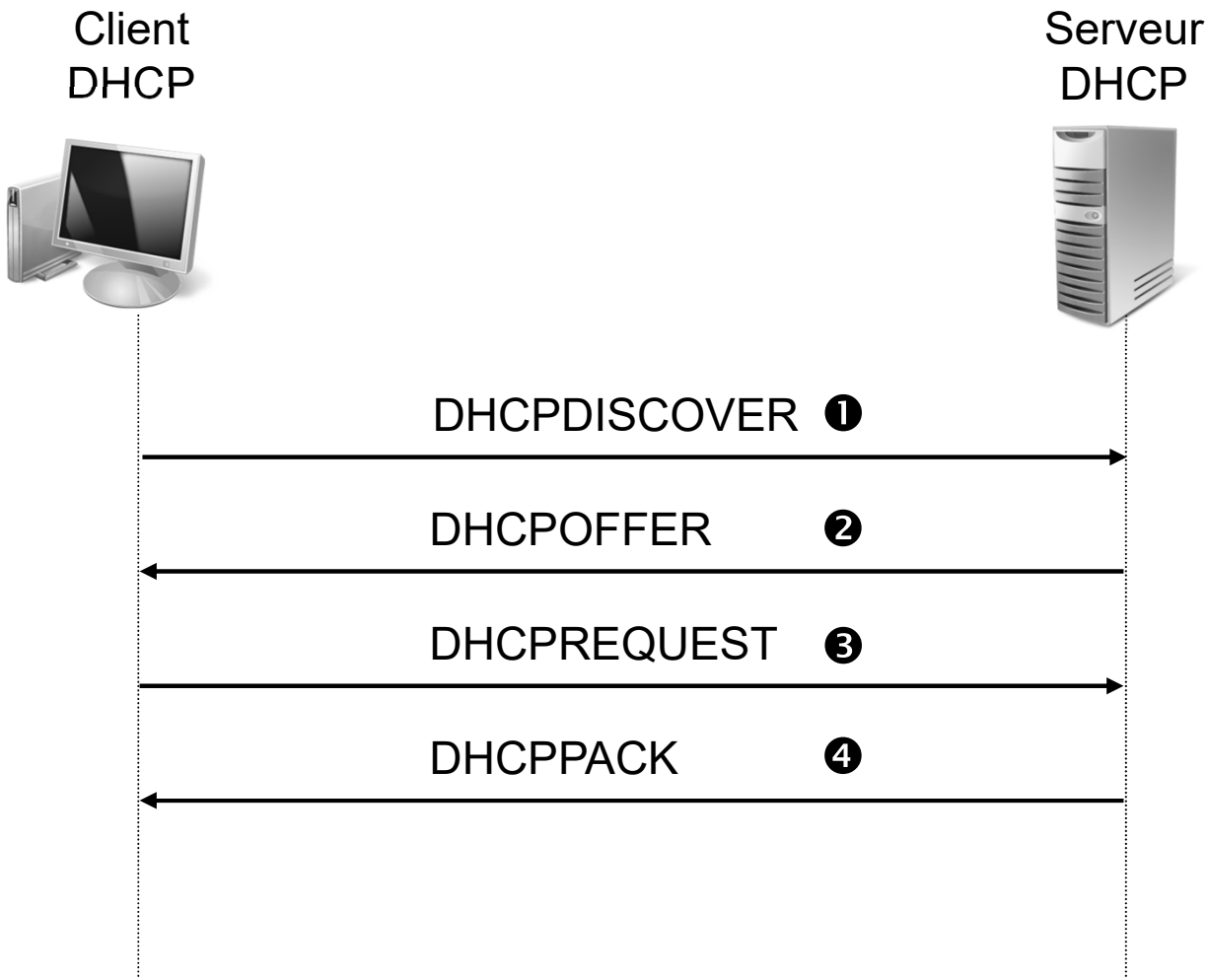
SSH (*Secure SHell*)

- Shell sécurisé
  - Destiné à remplacer les commandes **`rlogin`** (ou **`telnet`**), **`rsh`** et **`rcp`**
  - Chiffre les communications entre le client et le serveur
  - Permet d'identifier les utilisateurs et machines en présence à l'aide de clés
- Utilise une connexion TCP
  - Utilise le port 22 pour le serveur
- Modes
  - client : processus d'un utilisateur
  - serveur : démon sshd
- Le client ouvre la connexion

DHCP (*Dynamic Host Configuration Protocol*)

- Défini par la RFC 2131
- Conçu comme une extension du protocole BOOTP (*Bootstrap Protocol*)
- Protocole fonctionnant en client/serveur
- S'appuie sur UDP (ports 67 et 68)
- Permet la configuration automatique des paramètres TCP/IP (adresse IP, masque, gateway …) des différents hôtes du réseau
  - 3 méthodes d'allocation des adresses IP
    - Allocation manuelle : attribution par le serveur DHCP d'une adresse IP définie par l'administrateur
    - Allocation automatique : attribution automatique par le serveur DHCP d'une adresse IP
    - Allocation dynamique : attribution par le serveur DHCP d'une adresse IP pour une certaine durée (bail)

Configuration d'un client DHCP



Configuration d'un client DHCP

❶ Le client (d'adresse IP inconnue 0.0.0.0) envoie une requête DHCPDISCOVER en broadcast (255.255.255.255) dans laquelle il insère son adresse MAC

❷ Le serveur lui répond avec un DHCPOFFER émis aussi en broadcast, qui contient

- L'adresse MAC du client
- La durée du bail
- L'adresse IP du serveur

❸ Si le client accepte, il envoie un DHCPREQUEST pour recevoir les paramètres

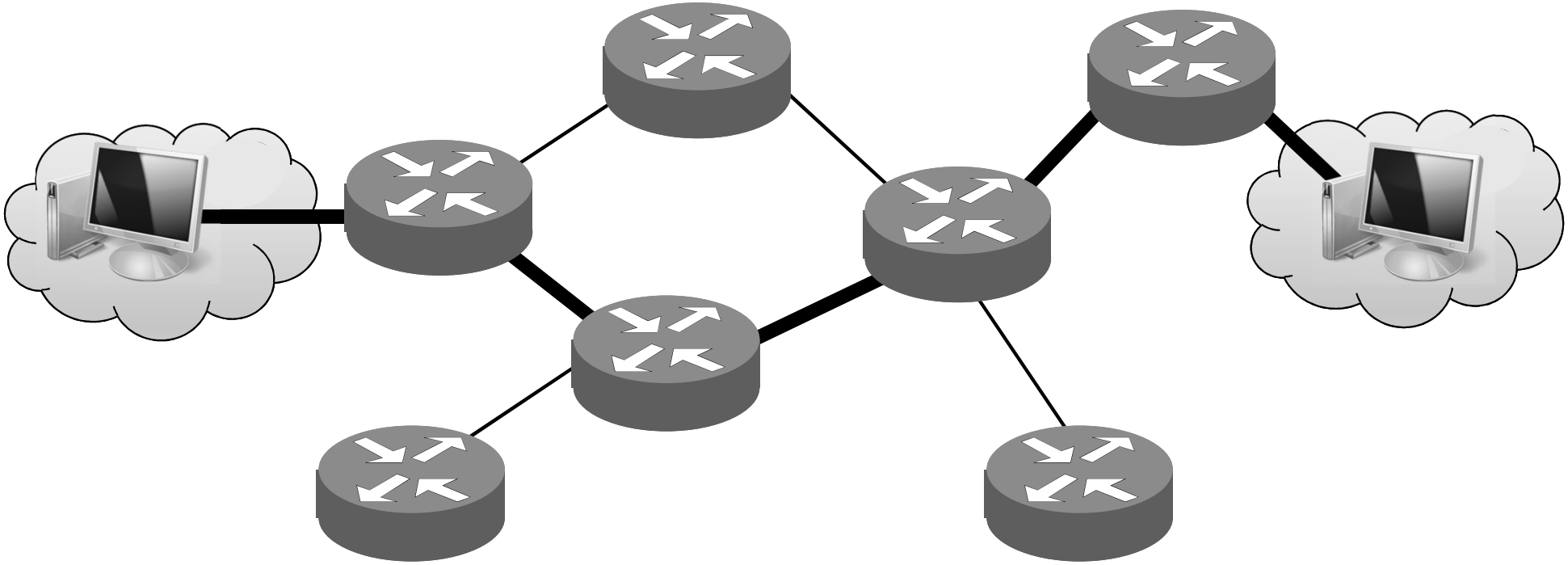
❹ Le serveur envoie un DHCPACK confirmant que le client accepte

Le routage IP

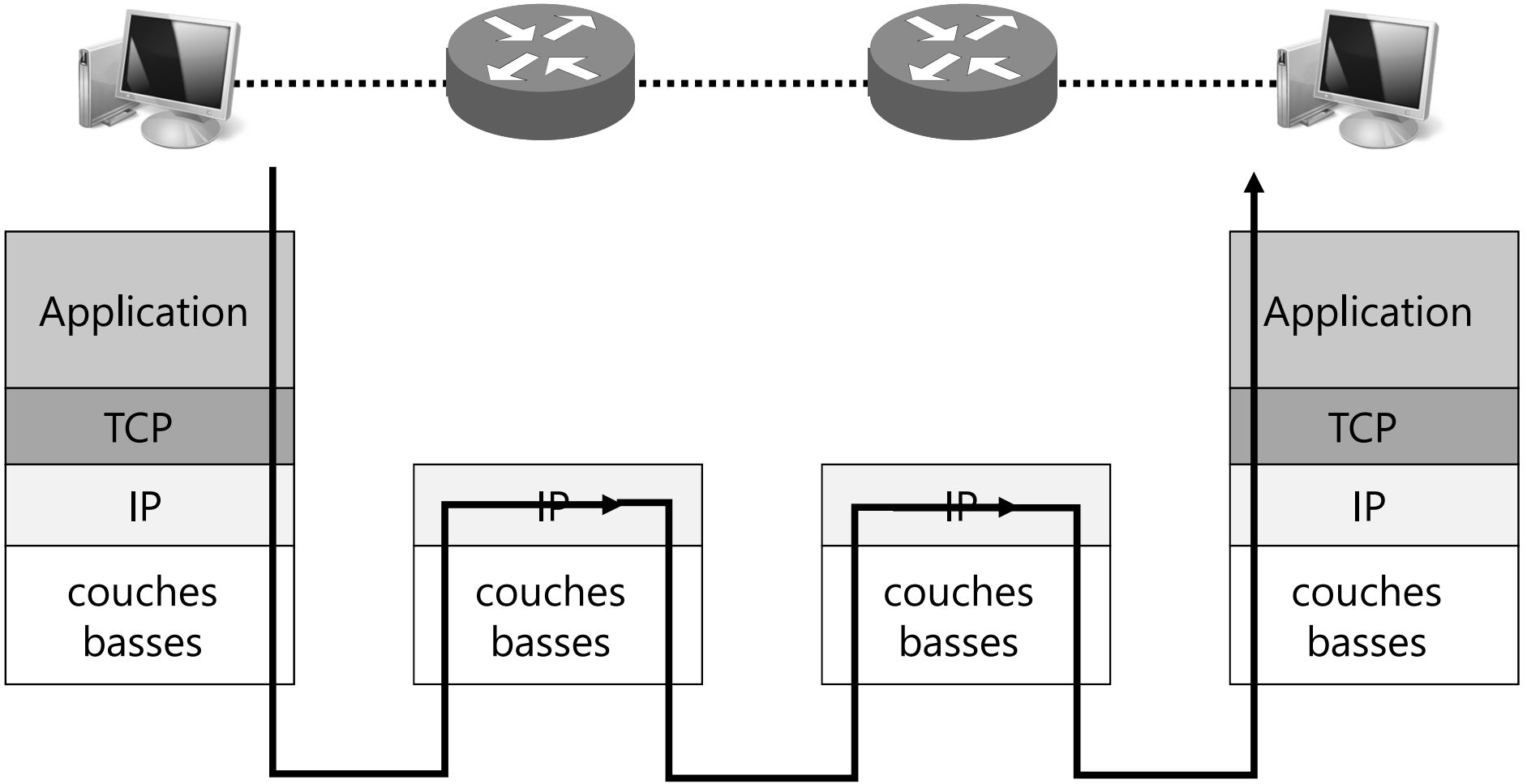
Les principes du routage IP

- Le protocole IP est capable de choisir un chemin (une route) suivant lequel les paquets de données sont relayés de proche en proche jusqu'au destinataire
- A chaque relais sur la route correspond un routeur (gateway)
  - L'ordinateur émetteur du paquet de données doit trouver le premier relais
  - Chaque routeur est chargé de trouver le suivant
  - Le dernier routeur remet le paquet sur le réseau du destinataire
- Le routage IP fonctionne de façon décentralisée : aucun nœud du réseau n'a une vision globale de la route que prendront les paquets de données

Les principes du routage IP



Cheminement du datagramme



Les tables de routage

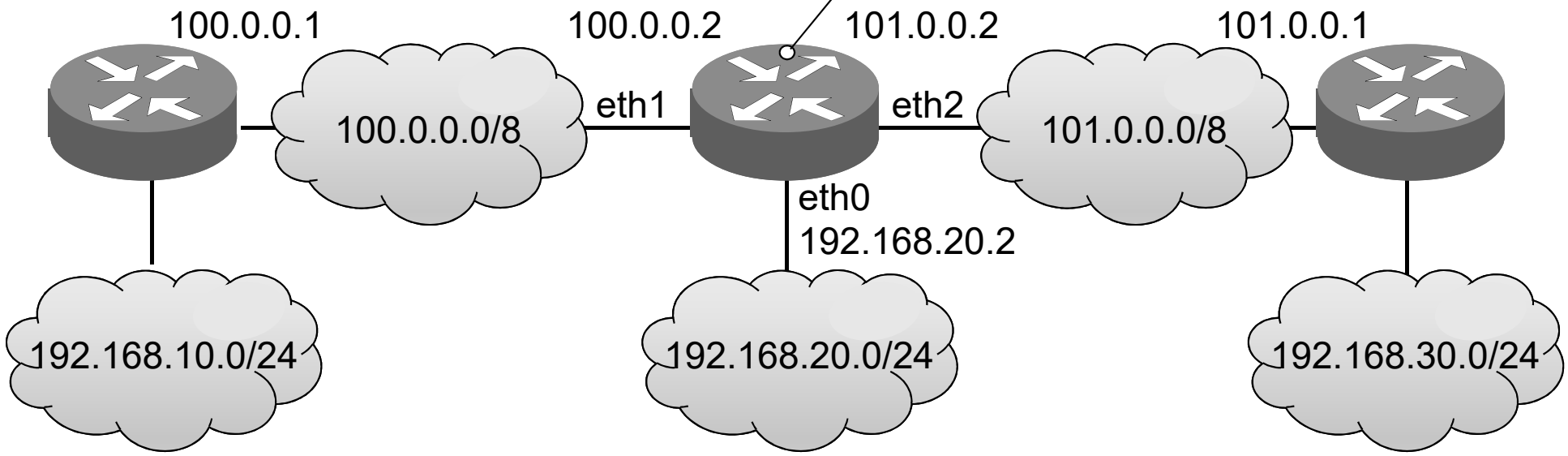
- Chaque équipement possède
  - une interface sur chaque réseau sur lequel il est connecté
    - sous Linux, ces interfaces portent les noms eth0, eth1 …
  - une table de routage qui contient essentiellement deux types d'information :
    - des adresses réseau
    - et le moyen de les atteindre
      - si le réseau est directement connecté à l'appareil, le moyen d'atteindre le réseau est le nom de l'interface
      - sinon il s'agit de l'adresse du routeur de prochain pas (« next hop ») situé sur la route vers ce réseau

Les tables de routage

- La table de routage est présente dans les **hôtes** comme dans les **routeurs**
  - un hôte ne traite que les paquets dont il est l'émetteur
  - un routeur traite tous les paquets reçus et dont il n'est pas l'émetteur
- La mise à jour de la table de routage peut être
  - manuelle : routage STATIQUE
  - automatique : routage DYNAMIQUE
- Accès à la table de routage
  - d'une station (UNIX, NT) : **`netstat -r[n]`**
  - d'un routeur (CISCO) : **`show ip route [sum]`**

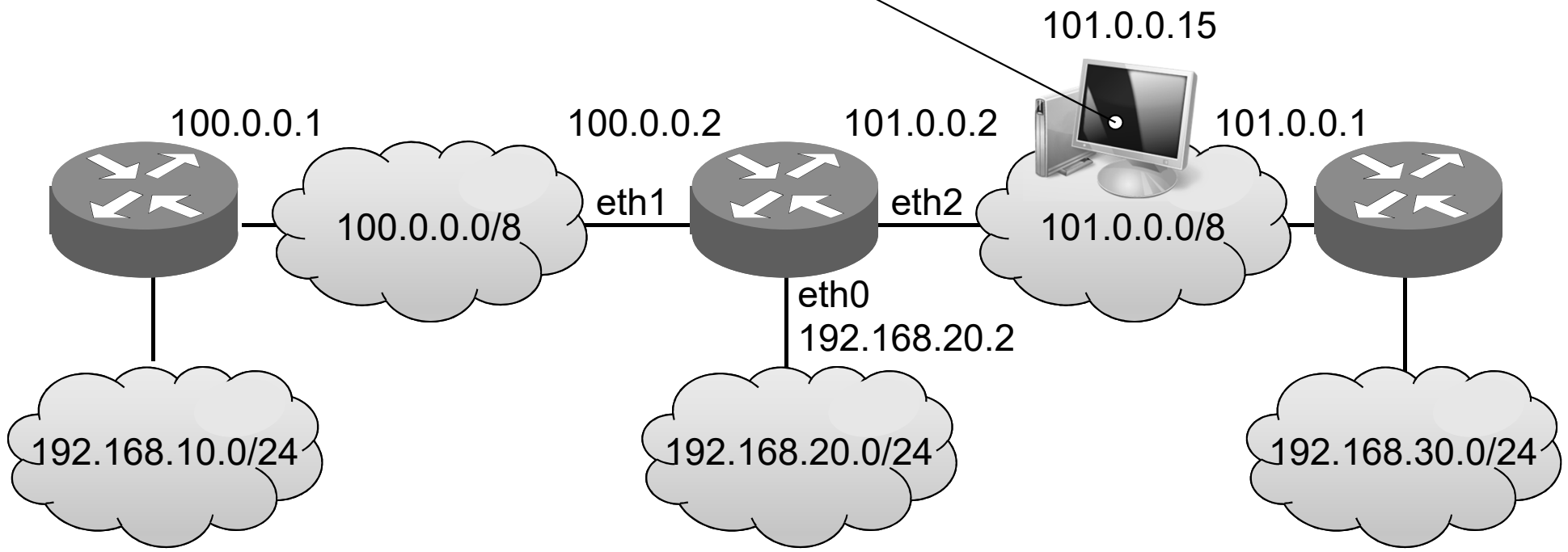
Les tables de routage

Destination	Moyen de l'atteindre
192.168.20.0/24	eth0
100.0.0.0/8	eth1
101.0.0.0/8	eth2
192.168.10.0/24	100.0.0.1
192.168.30.0/24	101.0.0.1



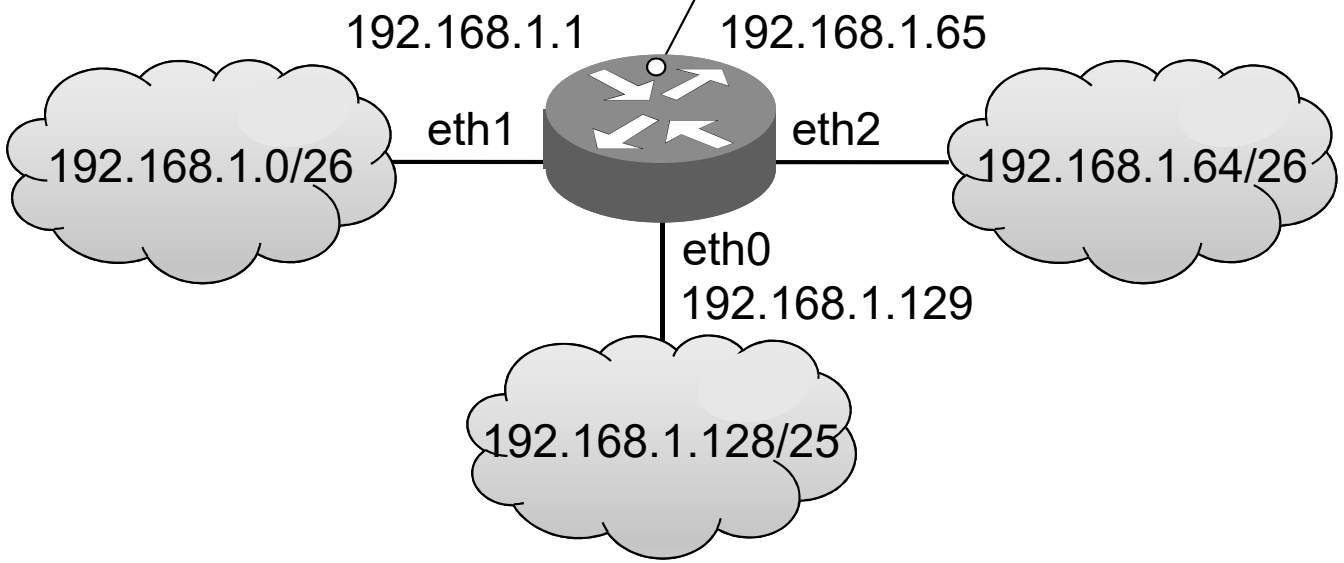
Les tables de routage

Destination	Moyen de l'atteindre
101.0.0.0/8	eth0
192.168.30.0/24	101.0.0.1
0.0.0.0	101.0.0.2



Routage entre sous-réseaux

Destination	Moyen de l'atteindre
192.168.1.128/25	eth0
192.168.1.0/26	eth1
192.168.1.64/26	eth2



L'algorithme de routage

- Algorithme exécuté lors de l'émission d'un paquet de données :
  - ❶ calcul du préfixe de réseau de l'adresse destination à l'aide du masque
  - ❷ recherche du préfixe dans la table de routage :
    - si le préfixe correspond à celui d'un réseau directement connecté, il y a **remise directe** du paquet sur le réseau (protocole ARP) et fin du routage
    - si le préfixe correspond à celui d'un réseau accessible via un routeur, le paquet est **transmis** au routeur concerné
    - si le préfixe n'a pas de correspondance dans la table, mais qu'il existe un routeur par défaut défini, le paquet est **transmis** au routeur par défaut
    - si aucun des cas précédents n'est rempli, une erreur de routage est déclarée

Les types de routage

- Routage STATIQUE
  - Les tables de routage des routeurs doivent être configurées manuellement par un administrateur réseau
  - Elles doivent être mise à jour manuellement à chaque fois qu'un changement intervient au niveau de la topologie du réseau
- Routage DYNAMIQUE
  - Après configuration pour démarrer le routage dynamique, les tables de routage des routeurs sont construites automatiquement
  - Les tables de routage sont mises à jour périodiquement en fonction des modifications apportées au réseau

Utilisation de TCP/IP

- Avantages
  - gratuit
  - ouvert
  - indépendant des constructeurs
  - disponibles sur tous les types de matériel : micro-ordinateur, station de travail, super calculateur et équipements réseaux
  - facile à installer
  - produits éprouvés depuis longtemps dans un monde hétérogène
  - inclut de très nombreuses applications
  - bien standardisé et documenté
  - les protocoles sont simples mais efficaces

Utilisation de TCP/IP

- Handicaps
  - les standards sont édités aux USA ; il n'y a pas de norme internationale
  - la plage d'adresses IPv4 est désormais épuisée
  - le protocole est très ouvert : on peut créer facilement un réseau que rapidement on ne peut plus gérer
  - il n'y a pas de routage basé sur l'adresse d'origine
  - la sécurité n'est pas prise en compte dans la conception ; de plus le mode non-connecté est un problème difficile pour la sécurité