R2.04 – Les bases des réseaux

RESPONSABLE : CRISTINA ONETE

MATERIEL : HTTPS://WWW.ONETE.NET/TEACHING.HTML

EMAIL : CRISTINA.ONETE@GMAIL.COM

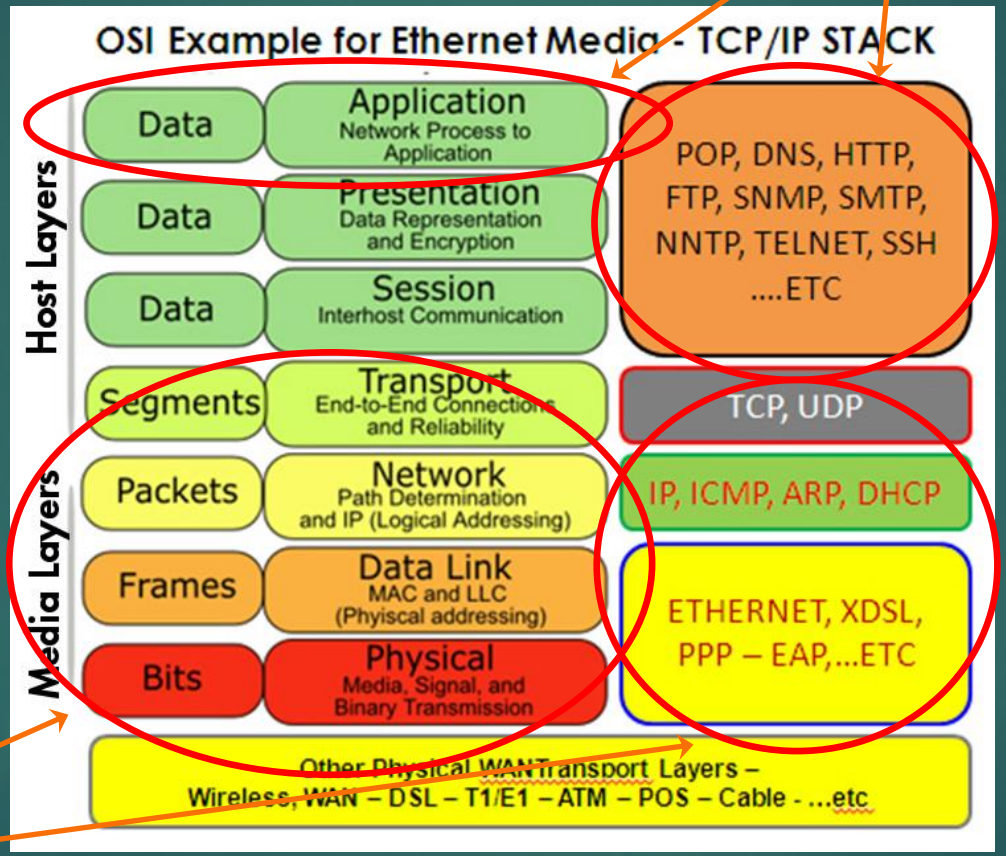
Les réseaux

- Partout dans nos activités quotidiennes :
  - Surfer l'Internet
  - Jouer aux jeux en réseaux
  - Parler aux amis : chat, email, sms...
- Même là où on s'imagine pas :
  - Téléphone & voitures communiquent toujours
  - En logistique et transports
  - Pour les systèms d'opérations
  - ...

R2.04, R2.05

R2.05

Les couches architecturales des réseaux



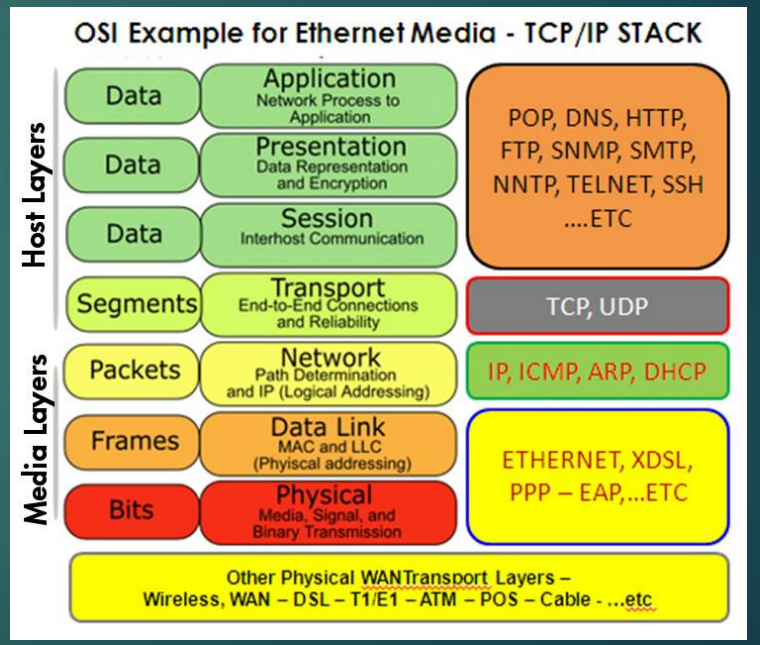
Les protocoles des réseaux

R2.04

Source : ipv6.com

Une autre vue : modèle TCP/IP

- Le modèle OSI se concentre sur des données
  - Modèle abstrait et universel (pour tout type de réseau)
- Une autre vue c'est le modèle TCP/IP
  - Modèle plus concrèt
  - Se concentre sur des protocoles plutôt que sur le format de données
  - Nous allons traiter une partie de ces protocoles

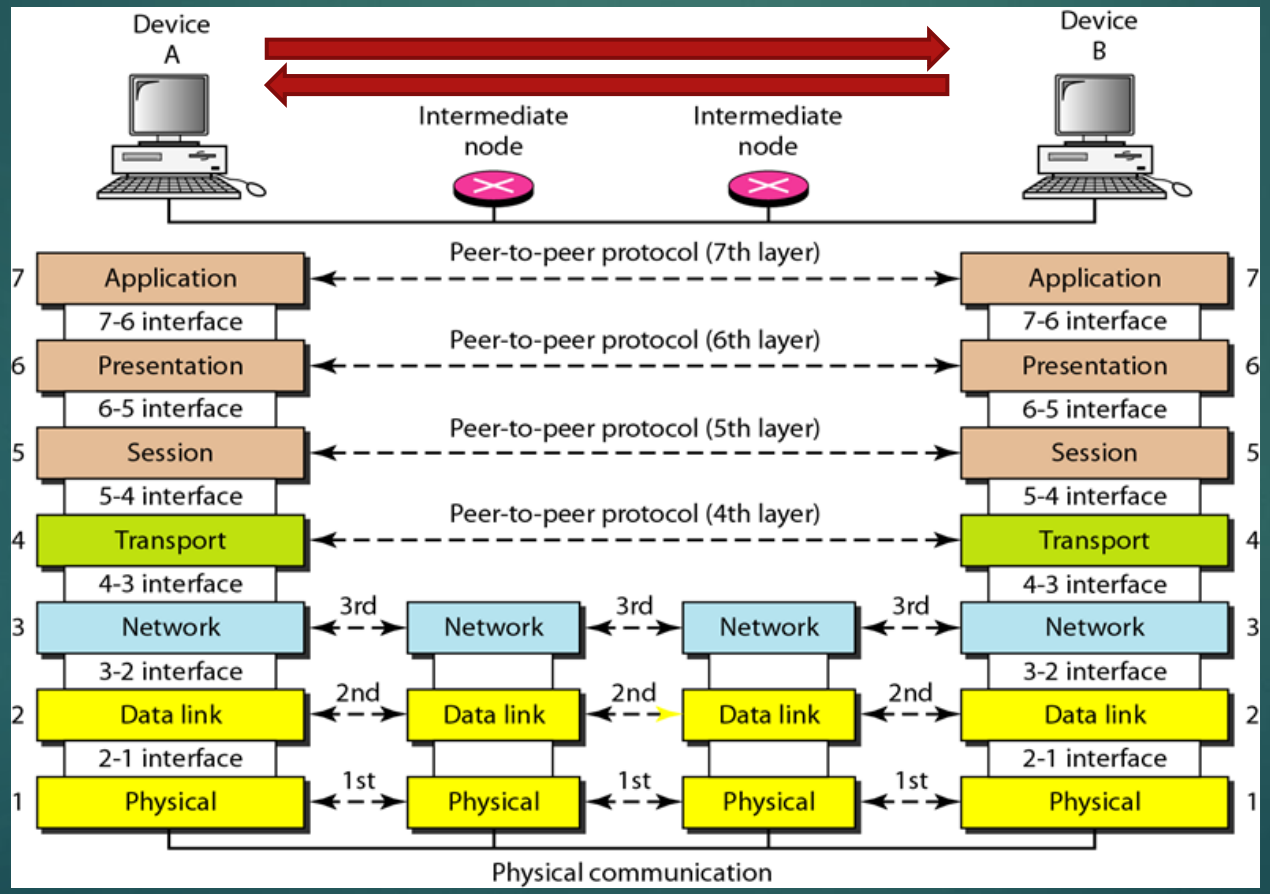


La structure des réseaux

LES MODELES OSI ET TCP/IP

Communication sur réseau

Ce qu'on voit
A contacte B



En réalité :

- des nœuds intermédiaires
- une infrastructure complexe

Les couches OSI

- OSI = Open Systems Interconnection
- Un modèle d'abstraction/ de communication en réseau :
  - décrit n'importe quelle réseau, y compris les réseaux informatiques
- 7 couches independentes :
  - Les couches basses – vers le medium physique
  - Les couches hautes – vers l'application
  - Chaque couche a un rôle bien défini
  - Le fonctionnement d'une couche ne dépend pas du tout de la structure des couches inférieures/supérieures

L'encapsulation

- En réseau, chaque couche doit être indépendente de l'autre
- Le traitement des messages à l'envoi/reception est fait couche par couche
- Cela s'appelle **l'encapsulation/decapsulation** du message

Envoi

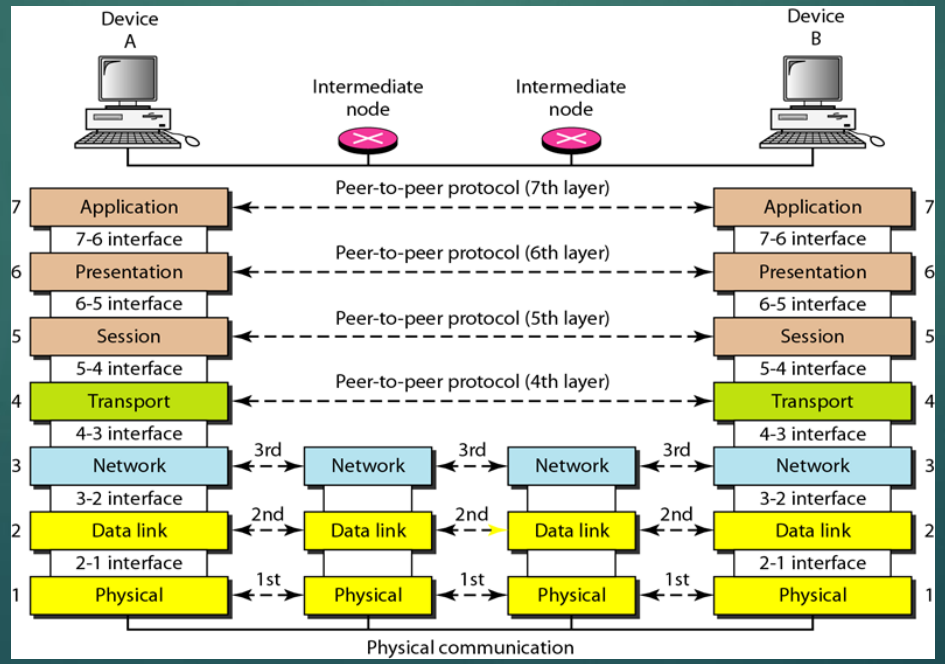
M_7 = message

M_6 = Traitement(M_7)

M_5 = Traitement(M_6)

...

M_1 = Traitement(M_2)



Réception

M_7 = Traitement(M_6)

M_6 = Traitement(M_5)

M_5 = Traitement(M_4)

...

M_2=Traitement (M_1)

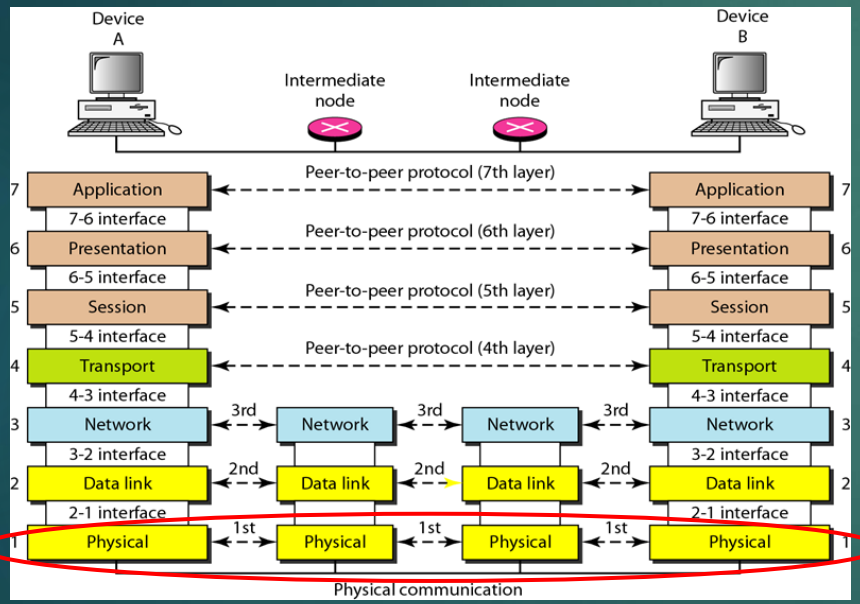
M_1

Un peu comme la poste...

- Indépendence des couches : exemple
  - Le message à envoyer = une enveloppe remise dans la poste
  - Le protocole d'envoi ne depend pas du contenu envoyé !
- On met donc ce qu'il faut envoyer dans une enveloppe et on écrit une adresse
  - Ceci est comme une encapsulation : même procédée pour toute entrée
- La poste livre l'envoi selon le même procédée, quelque soit le contenu

La couche physique

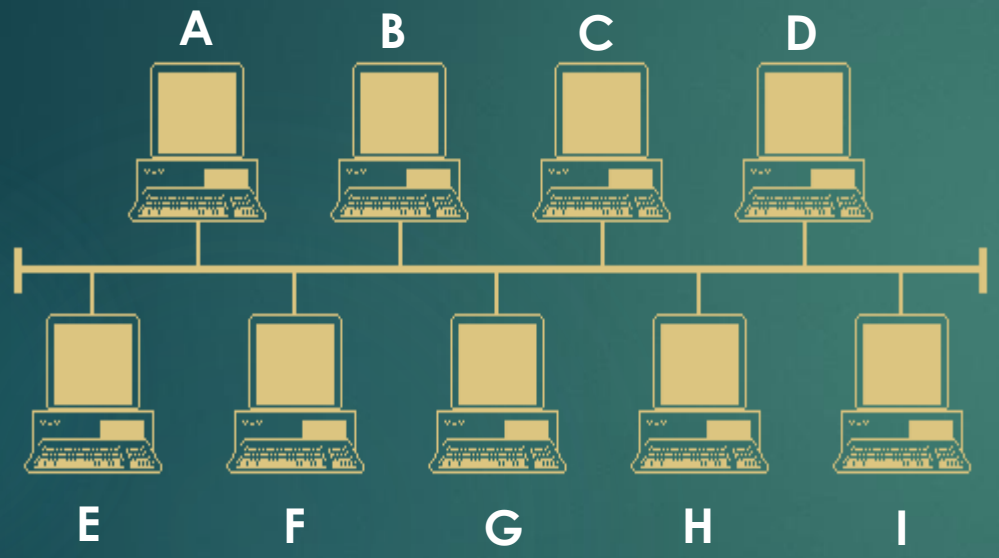
Description de la couche physique



- Toute communication passe finalement par un medium physique
- Communication possible seulement si une connexion physique directe existe
- Types de couches physiques :
  - Cable (cuivre) : courant = 1; aucun courant = 0
  - Fibre : lumière = 1; pas de lumière = 0
  - Des ondes radio (connexion sans fil)

Nous ferons très peu à cette couche dans ce cours

Medium partagé



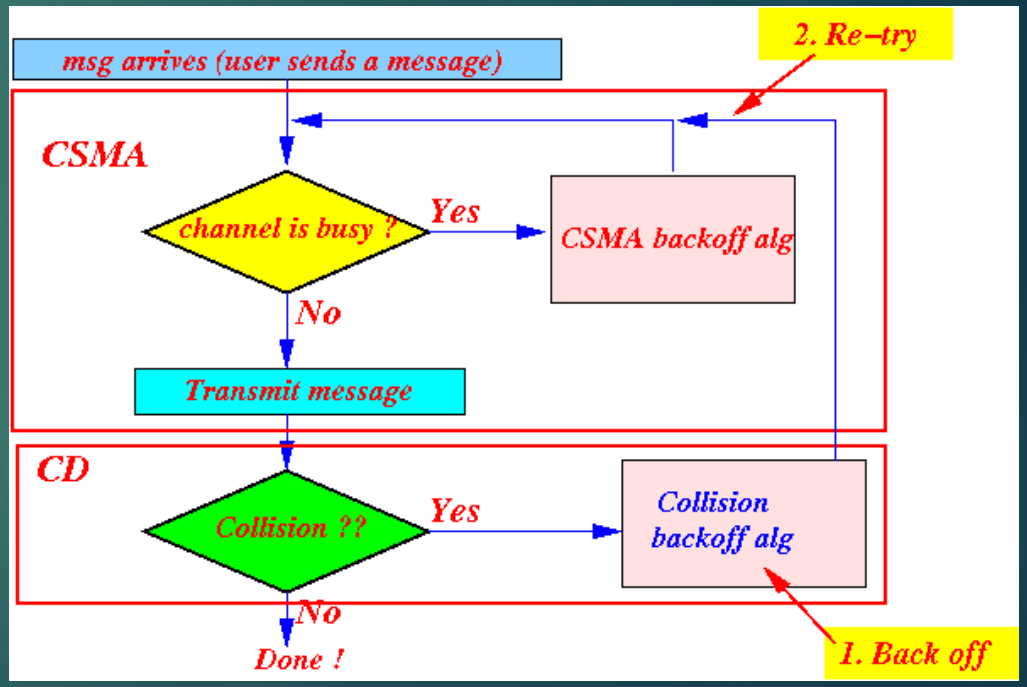
Ces ordinateurs partagent un medium physique

Des propriétés souhaitées :

- Détection des collisions de signaux
- Détection d'un signal déjà existent
- Transmission varidirectionnelle

Le protocole CSMA/CD

- = "Carrier sense multiple access/Collision Detection"
  - Carrier sense : l'émmeteur écoute avant d'envoyer, pour détecter des transmissions
  - Multiple access : medium partagé par des des divers périphériques
  - Collision detection : dès qu'une collision est détectée, la transmission s'arrête pour une ré-transmission



Eléments de topologie

- Au delà des cables typiques : cuivre, fibre, ...
- Il y a également des connecteurs :
  - Hub ou Switch

- Topologies typiques :

bus topology

haute probabilité de collisions

star topology

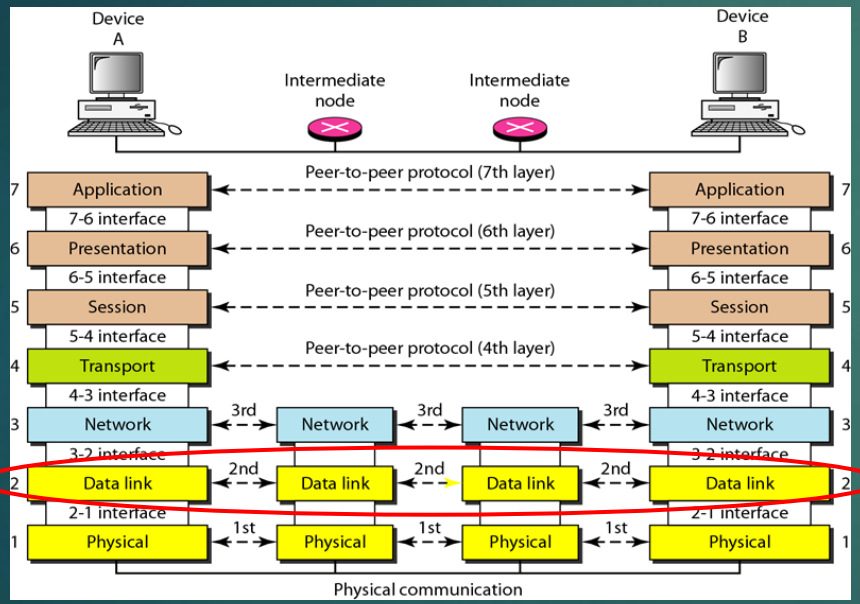
Hub vs. switch

- Hub : le broadcast
  - Un message envoyé par un hub arrive à tous ceux qui y sont connectés
  - Très utile aux broadcasts
  - Beaucoup d'overhead inutile en cas d'unicast

- Switch : le unicast
  - Message envoyé à des destinataires précis
  - Utile aux unicasts
  - Inefficace pour les broadcasts

La couche liaison

La couche liaison



La couche liaison

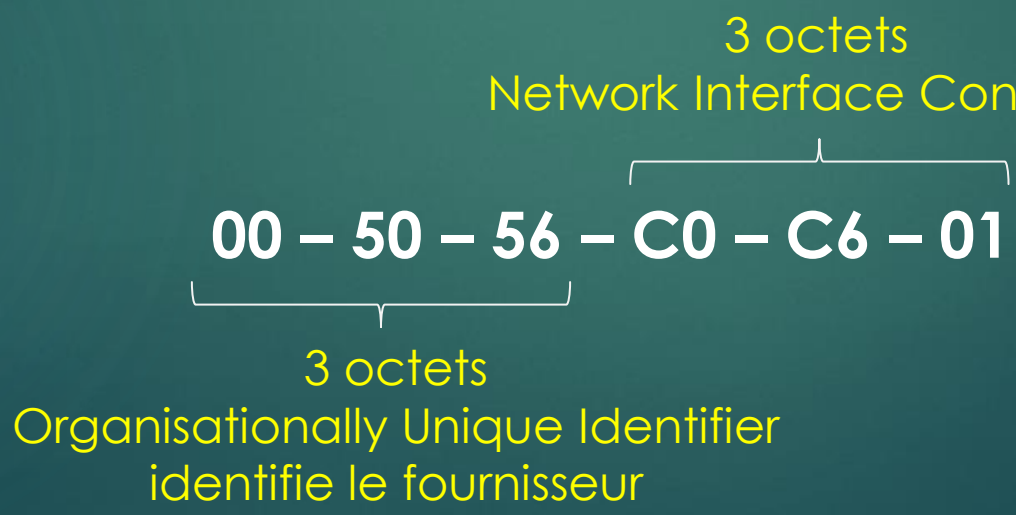
- S'applique encore seulement aux périphériques physiquement connectés
- **Objectif** : gérer la communication simultanée par des divers périphériques sur le même medium physique
- Signaux (bits) → trames
- Protocoles de transmission simultanée :
  - Par exemple Ethernet

À cette couche : transmissions intra-réseau, protocole ARP, @MAC

Les adresses MAC

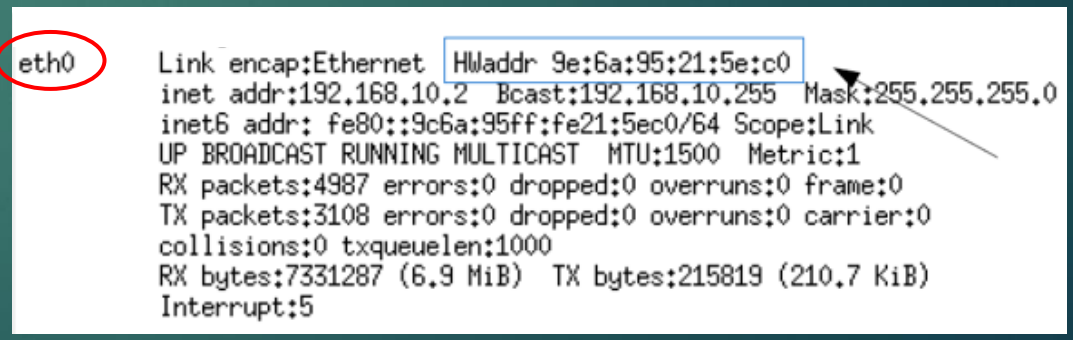
- Communiquer à la couche liaison : comment adresser des messages ?
- Les adresses MAC :
  - 48 bits = 6 octets en deux parties
  - unique et permanente
- Adresses spéciales : FF : FF : FF : FF : FF : FF pour broadcast

  33: 33 : xx : xx : xx : xx pour multicast



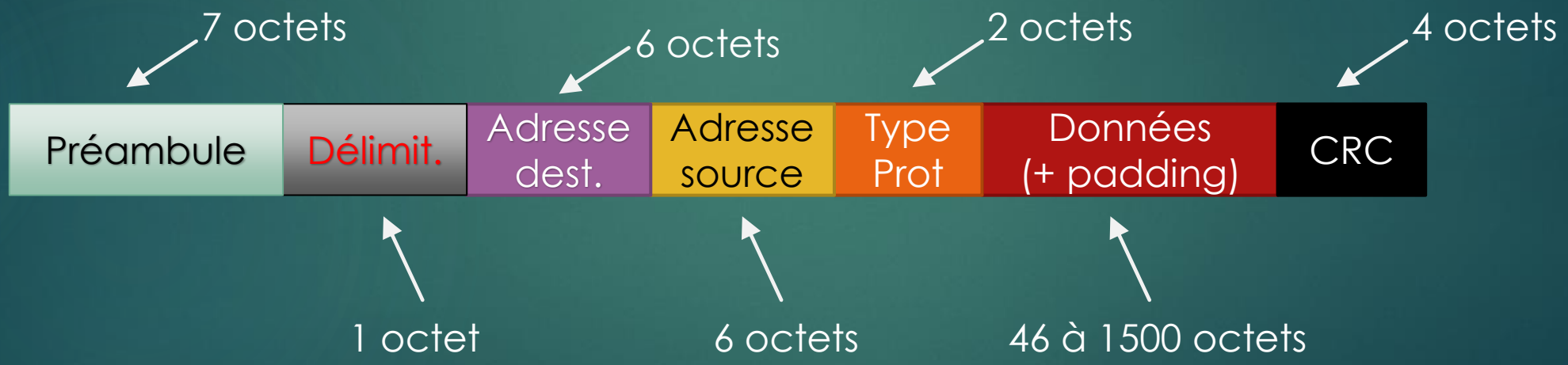
Trouver les adresses MAC

- Communication en réseau : interfaces réseau
  - Des identifiants numériques correspondants à un objet physique (carte réseau)
- Chaque interface aura sa propre adresse MAC (et IP)
- En Windows : ipconfig
- Dans les vieilles distributions Linux, la commande `ip address show` montrera les adresses utilisées:
  - L'adresse MAC : HWaddr



La trame Ethernet II

- Données : entre 46 et 1500 octets
  - du padding (bourrage) est nécessaire si taille inférieure



Hello world : couche liaison

PC 1
c2 : 90 : dc : 8f : 5a : ea

PC1 → PC2 : "bonjour"

PC 2
72 : 01 : ac : dc : f1 : 61

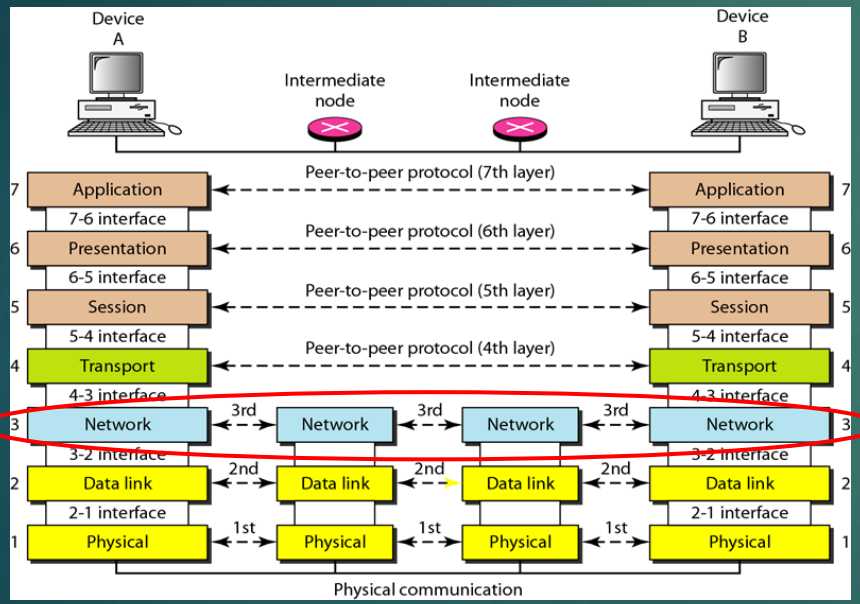
```
▸ Frame 1: 48 bytes on wire (384 bits), 48 bytes captured (384 bits)
▾ Ethernet II, Src: c2:90:dc:8f:5a:ea (c2:90:dc:8f:5a:ea), Dst: 72:01:ac:dc:f1:61 (72:01:ac:dc:f1:61)
  ▸ Destination: 72:01:ac:dc:f1:61 (72:01:ac:dc:f1:61)
  ▸ Source: c2:90:dc:8f:5a:ea (c2:90:dc:8f:5a:ea)
    Type: Unknown (0x2000)
▾ Data (34 bytes)
    Data: 626f6e6a6f7572010101010101010101010101010101010101...
    [Length: 34]
```

Protocoles couche 2

- C'est quoi un protocole "de couche 2" ?
  - Déjà, à cause de l'encapsulation, un protocole de couche 1 également !
- Protocole avec encapsulation couches 1 et 2, mais pas couche 3 !
  - Par exemple, le protocole ARP (qu'on verra très bientôt !)
- Le protocole marchera toujours au sein d'un seul réseau, sans routeur
  - Car la couche 2 fait la connexion intra-réseau

La couche réseau

La couche réseau



La couche réseau

- Interopérabilité entre les media physiques
- Couche liaison : communication même medium
- Couche réseau : la communication sur deux media physiques différents
- Par exemple : communiquer avec un voisin/ami
  - Notre réseau = 1 médium physique
  - Son réseau = 1 autre médium physique
  - Transmission demande des adresses réseau (IP)

À cette couche : transmissions inter-réseau, routage, @MAC, protocole IPv4

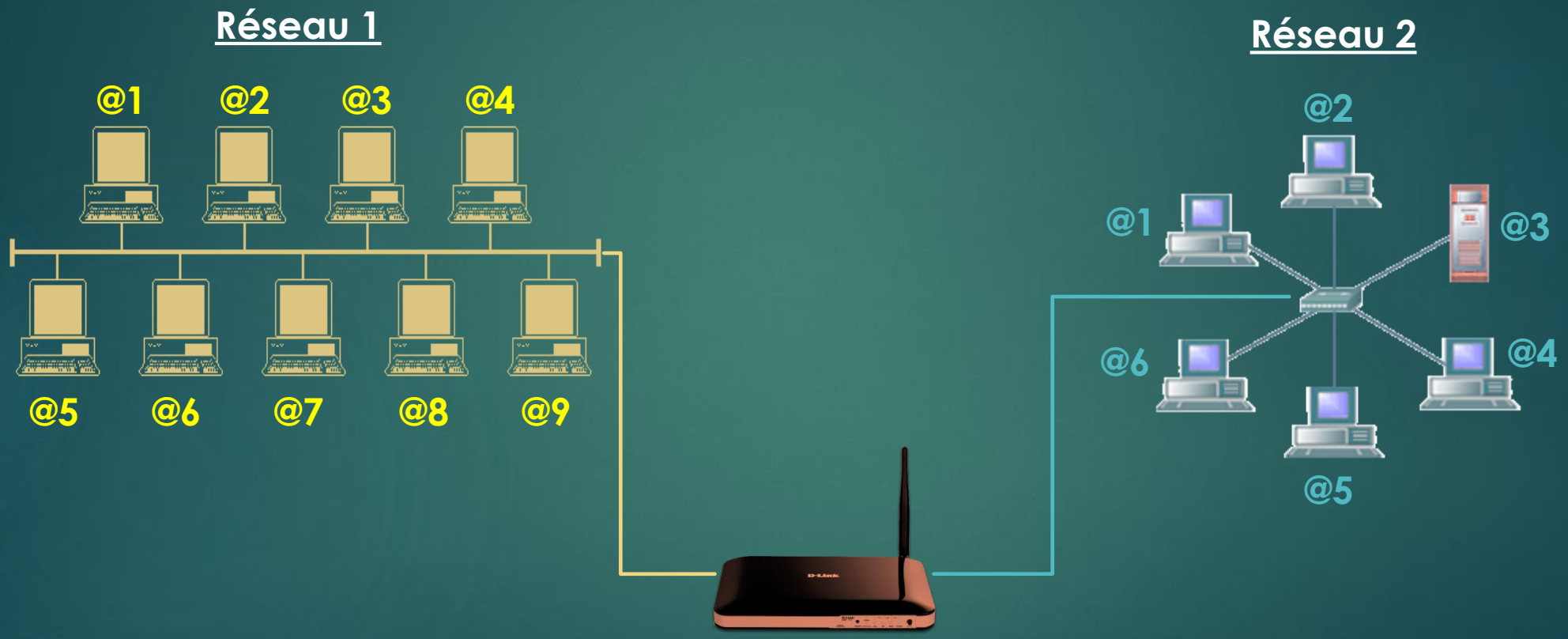
La problématique

Comment envoyer des messages entre les deux réseaux ?

Envoyer des messages : couche 2

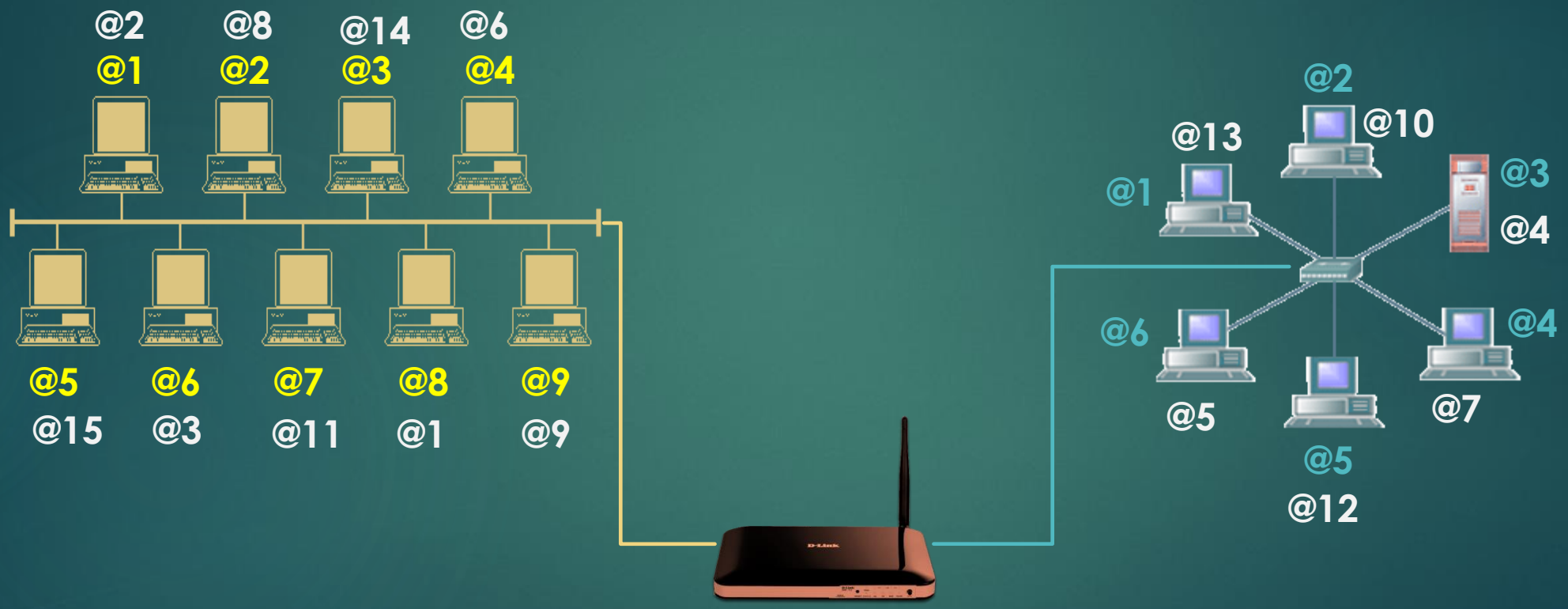
Envoyer des messages : couche 2

Une passerelle entre 2 réseaux



Passerelle – ici, un router

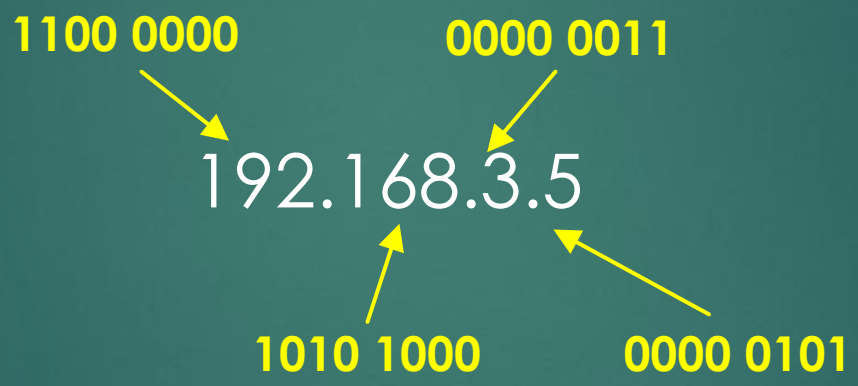
Les adresses IP



- La passerelle doit pouvoir associer un ordinateur à une adresse unique : **les adresses IP**

Adresse IP, domaine de collision

- Un numéro à 32 bits = 4 octets
  - Partagé en 4 groupes d'un octet chacun, séparés par des points



- Adresse MAC permanente & unique vs Adresse IP temporaire & non-unique
- Adresses IP avec un préfixe constant peuvent former des sous-réseaux
  - également appelées des domaines de collision

Domaine de collision, CIDR

- Domaine de collision = suite d'adresses, souvent contigües, même préfixe
- La taille du préfixe donne la taille du réseau
  - Préfixe à x bits (sur 32) => taille du réseau sera de $2^{32-x} - 2$
  - Pourquoi ?

    Une adresse IP a 4 octets = 32 bits. Si x bits constants (préfixe) dans un sousréseau, il restent 32-x bits sur lesquels on a les adresses machine. Un bit est soit 0 soit 1, d'où $2^{32-x} - 1$ possibilités. L'autre @ prise : broadcast

- On indique la taille d'un DC par la notation CIDR : 192.168.5.23 / 24
  - Ceci veut dire, le préfixe est sur 24 bits, il y a $2^8 - 2 = 254$ machines
  - Autre notation : masque de réseau : 24 fois 1, suivi par 8 fois 0 : 255.255.255.0

Structure d'un DC

- Prenons l'adresse machine 192.168.5.23/24
- **Masque de réseau** : 111111111. 11111111. 11111111. 00000000 = 255.255.255.0
- Partie réseau : les 24 premiers bits : 192.168.5
  - Adresse réseau : préfixe suivi par des 0s et CIDR : 192.168.5.0/24

- Partie machine :
  - Dernière adresse : @ broadcast
    - On prend l'@ réseau, on complète avec des 1s : 192.168.5.255
  - Première adresse machine : 192.168.5.1
  - Dernière adresse machine : celle d'avant le broadcast : 192.168.5.254

La taille d'un sous-réseau

- Problématique inverse :
  - Besoin d'une sous-plage qui accueille 75 machines. Quelle taille CIDR ?
  - CIDR sur 26 bits ? $2^6 - 2 = 64$ machines -- trop peu !
  - CIDR sur 25 bits : $2^7 - 2 = 126$ machines -- parfait !
  - Typiquement le CD formera une sous-plage d'une plage achetée

- Découper une plage : 12.23.128.0/18 pour arriver à une plage /19 :
  - Identifier le préfixe réseau : 18 bits = 2 octets + 2 bits : **12.23.10**000000.0
  - Découper la plage en 2 parties disjointes : jouer sur le 19ème bit :
    - Sous-réseau 1 : **12.23.100**00000.0/19
    - Sous-réseau 2 : **12.23.101**00000.0/19

Configuration : commande ip

- Exemple : l'adresse 192.168.3.129 dans le réseau 192.168.3.128/25

- Éléments nécessaires : @IP, masque CIDR, interface à configurer
  - On indique à l'interface : son @IP & le DC dans lequel elle se trouve
  - Deux méthodes :

```
ip a add 192.168.3.129/25 dev eth0
```

Méthode 2 : fichier /etc/network/interfaces (sur la majorité des distributions)

On verra cette méthode plus tard

Les adresses réseaux

- Vérifier qu'une adresse est correcte :
  - Une adresse IPv4 n'a que 4 octets :
    - Chaque octet DOIT prendre une valeur entre 0 et 255
    - 23.15.289.1 n'est pas une adresse IPv4 valide !
  - Une adresse réseau IPv4 DOIT ne contenir que des 0s pour la partie machine !
    - 122.12.65.0/23 n'est pas une adresse valide !
      - ... car 65 = [0100 0001] en binaire.
      - ... et /23 veut dire **122.12.[0100 0001]**.0 partie machine
      - ... avec un 1 dans la partie machine!!

Un autre système, plus ancien

- CIDR = **classless** inter-domain routing
- Le système de classes est encore en usage, même s'il est désuet
  - Particulièrement utilisé dans les configurations des machines
  - Indique le taille par défaut d'un réseau en fonction de ses premiers bits

Classe	Bits start	Bits # réseau	Bits # machine	Max # réseaux	Max # machines	Adresse début	Adresse fin	CIDR
A	0	8	24	2^7	2^{24}	0.0.0.0	127.255.255.255	/8
B	10	16	16	2^{14}	2^{16}	128.0.0.0	191.255.255.255	/16
C	110	24	8	2^{21}	2^8	192.0.0.0	223.255.255.255	/24
D	1110	N/A	N/A	N/A	N/A	224.0.0.0	239.255.255.255	N/A

Adresses spéciales

- Adresse de broadcast : par défaut la dernière de la plage
- Adresse de loopback : 127.0.0.1
- Adresses privées :
  - Non-routables
  - Peuvent se répéter dans des domaines de collision différents
  - Plages : **10.0.0.0/8**

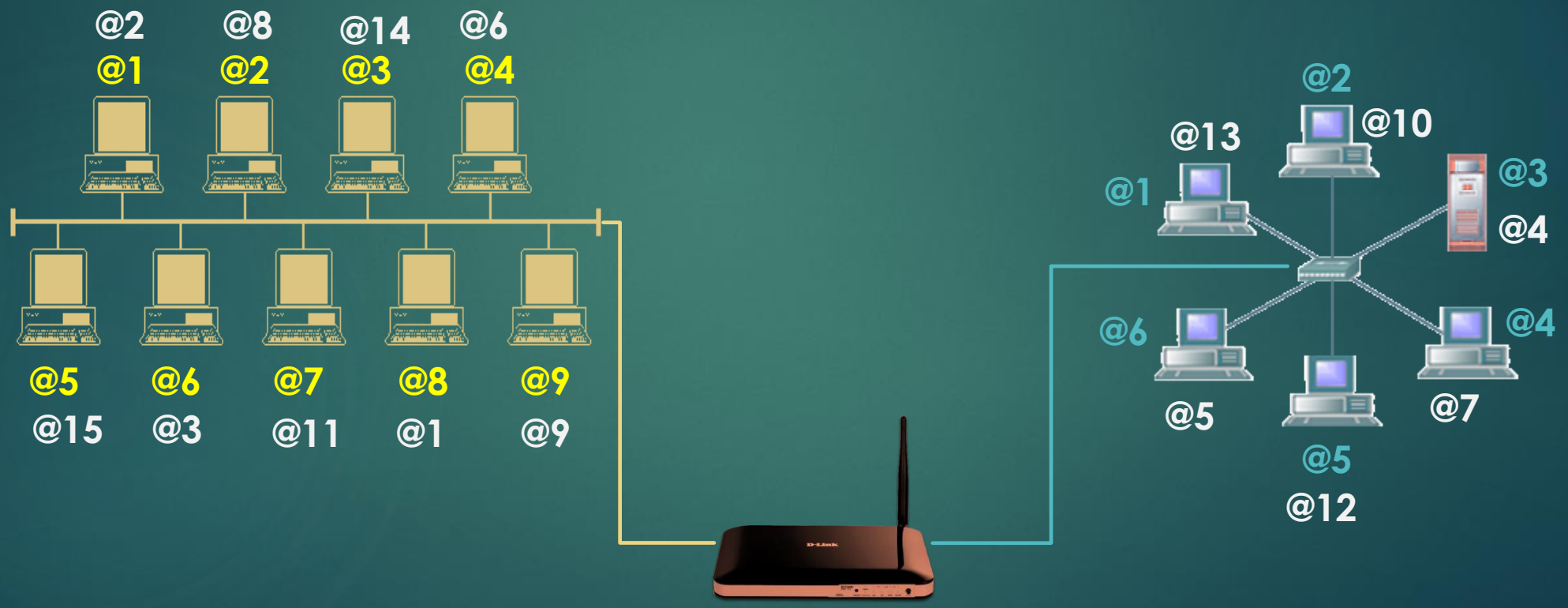
    172.16.0.0/12

    192.168.0.0/16
  - Les plages privées sont utilisées pour s'adapter au nombre croissant de périphériques connectés

Couche réseau : envoyer un paquet

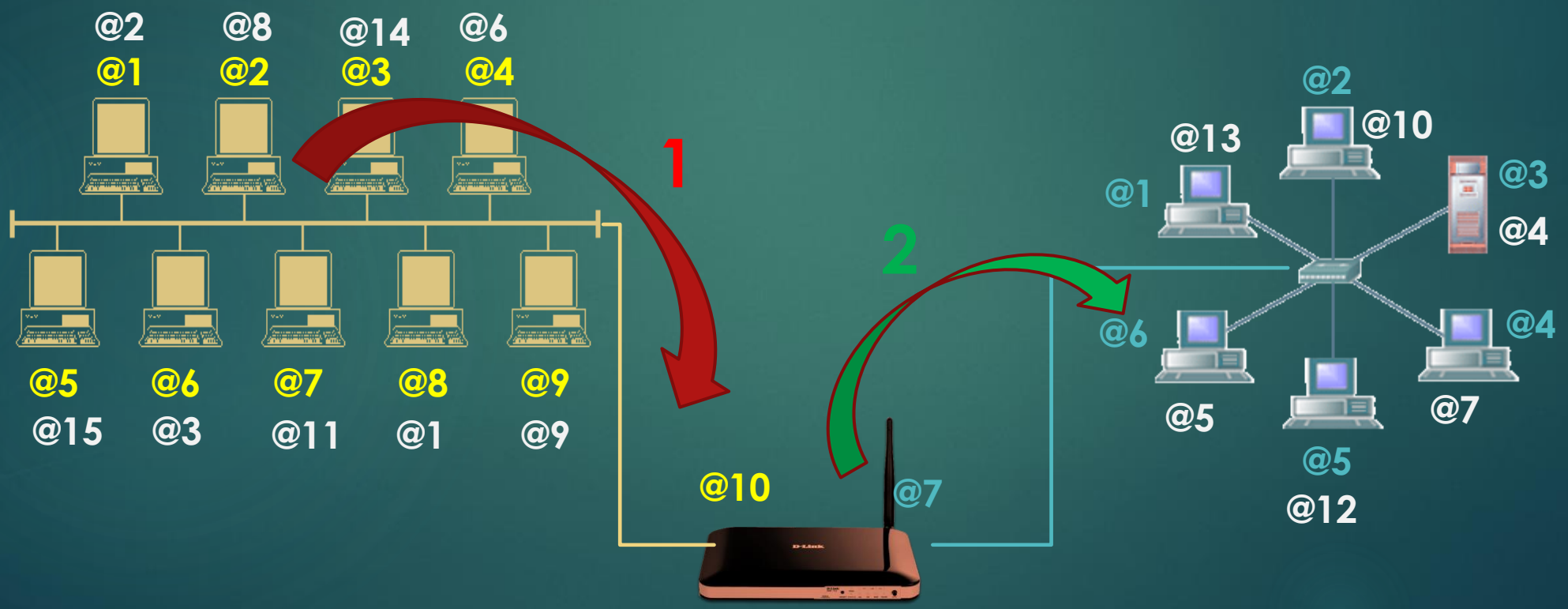
Envoi inter-réseau

- Problème : comment disons @2 (**@8**) peut-il envoyer un message à @6 (**@5**) ?



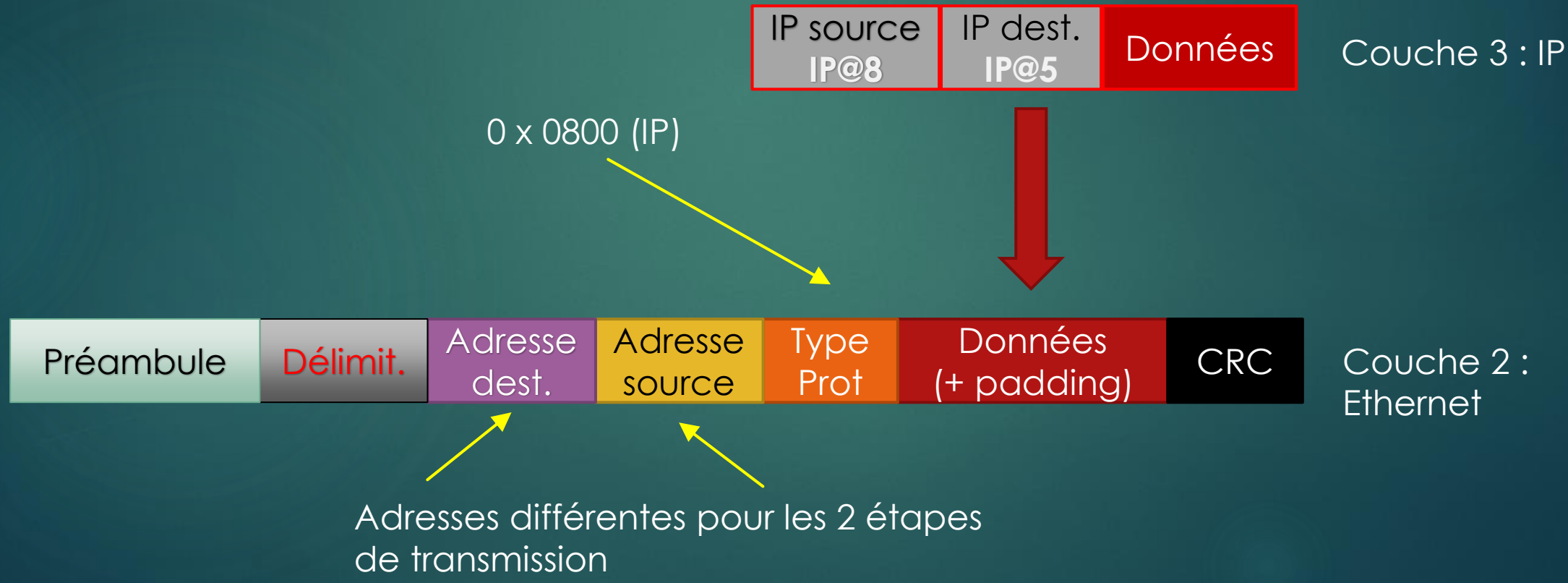
Envoi inter-réseau

- Problème 2 : comment disons @2 (**@8**) peut-il envoyer un message à @6 (**@5**) ?



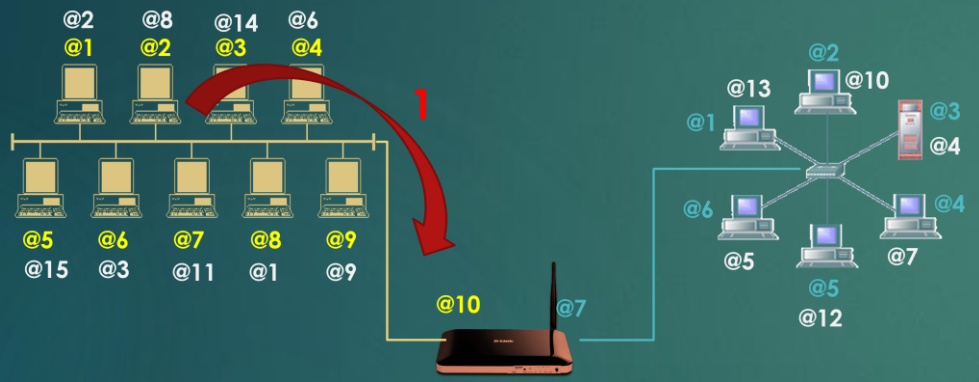
Transmission inter-réseaux

- Deux étapes de transmission, les deux intra-réseau
- Utilisation du protocole IP pour l'encapsulation à la couche 3



Transmission inter-réseaux

- Première étape de transmission : @8 → passerelle



IP source IP@8	IP dest. IP@5	Données

Couche 3 : IP

Préambule	Délimit.	Adresse dest.	Adresse source	Type Prot	Données (+ padding)	CRC

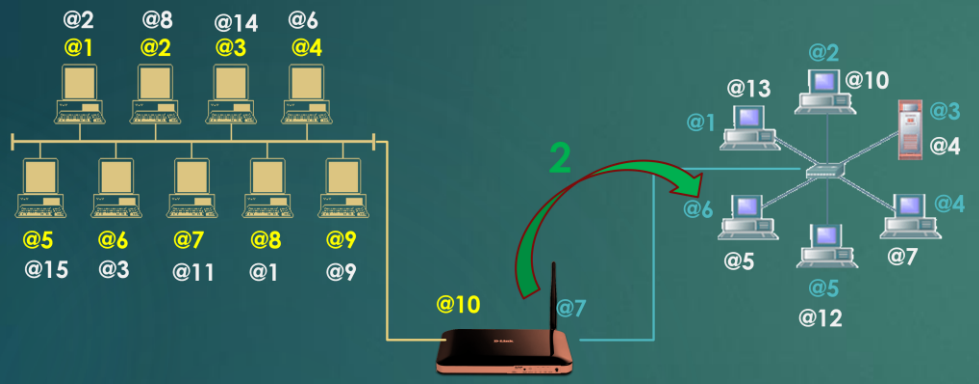
Couche 2 : Ethernet

@MAC : passerelle, côté gauche

@MAC machine @8

Transmission inter-réseaux

- Première étape de transmission : @8 → passerelle



IP source IP@8	IP dest. IP@5	Données

Couche 3 : IP

Préambule	Délimit.	Adresse dest.	Adresse source	Type Prot	Données (+ padding)	CRC

Couche 2 : Ethernet

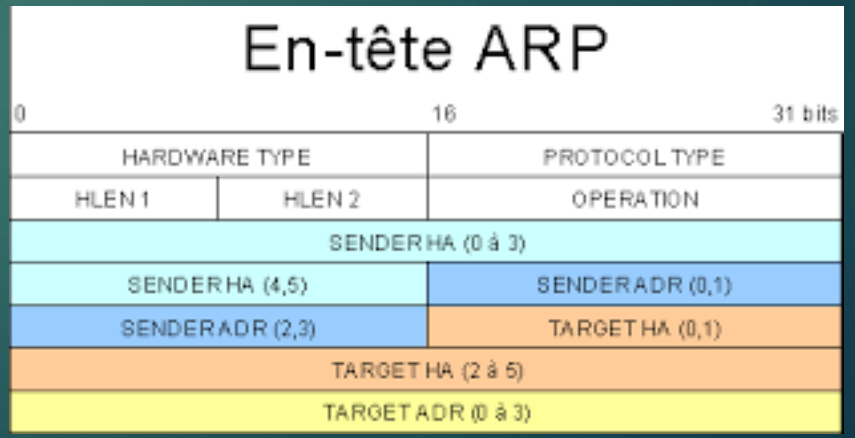
@MAC machine @5

@MAC passerelle côté droit

Association adresses MAC/IP

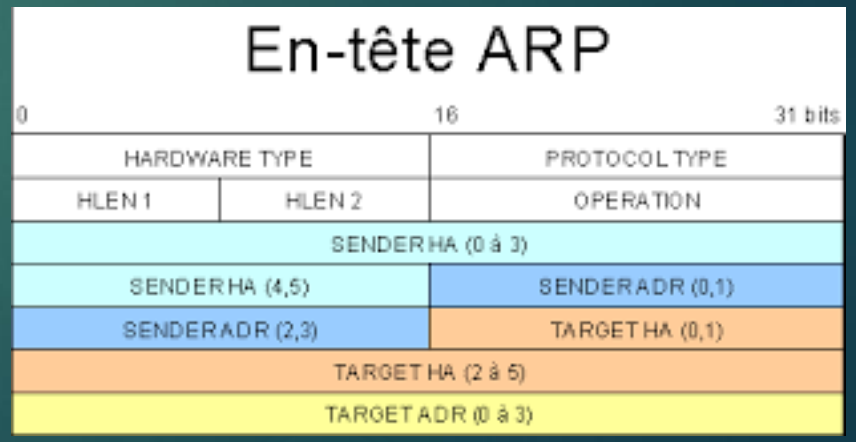
- Comment savoir quelle adresse MAC correspond à quelle adresse IP ?
  - Protocole ARP, couche 2
  - Ne peut s'exécuter que dans un seul réseau (DC)
  - Exécuté par une machine cherchant une @ MAC à partir d'une @IP et une machine qui connaît l'@MAC cherchée

- ARP : 2 types de messages : requête, réponse, encapsulées en Ethernet
  - En-tête spécifique ARP
  - Chaque message ARP spécifie :
    - Machine src : @MAC, @IP
    - Machine dst : @MAC
    - Machine "cible" : @MAC, @IP



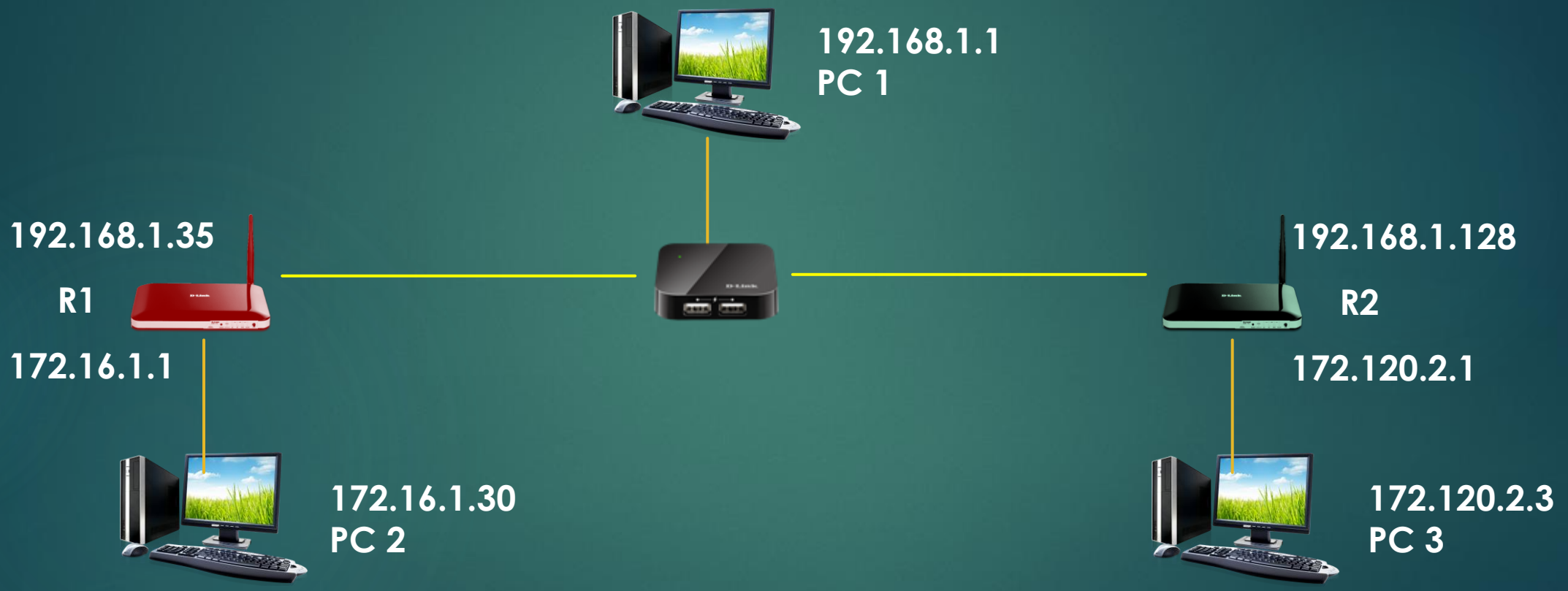
Association adresses MAC/IP

- Situation : Machine A (@IP ip_A @MAC mac_A) cherche @MAC de la machine B (ip_B)
  - Machine C (ip_C, mac_C) connaît la réponse
- ARP requête :
  - Machine src : MAC mac_A, IP ip_A
  - Machine dst : MAC broadcast : FF:FF:FF:FF:FF:FF : broadcast intra-réseau !
  - Machine cible : @MAC inconnue 00:00:00:00:00:00
- ARP réponse :
  - Machine src : MAC mac_C , IP ip_C
  - Machine dst : MAC mac_A
  - Machine cible : @MAC mac_B , IP ip_B



Couche réseau : le routage

Problématique -- le routage



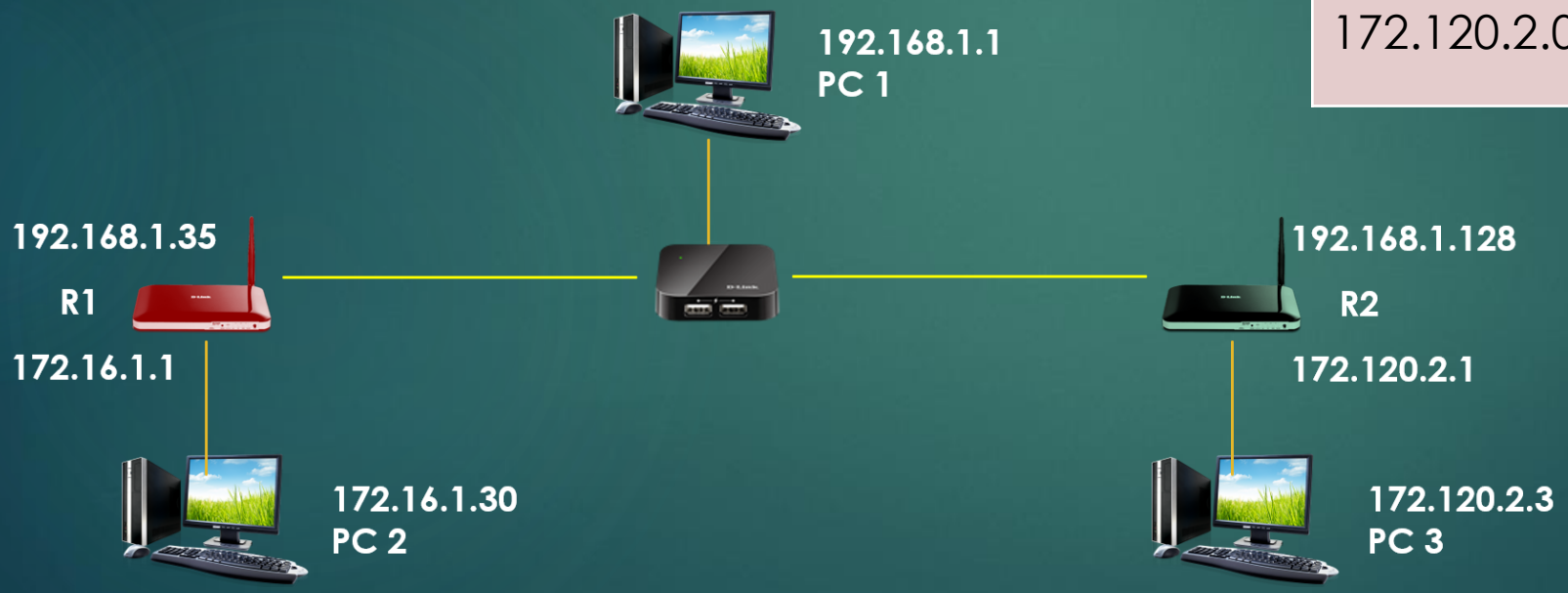
Comment PC 1 peut-il envoyer un message à PC 2 ou PC 3 ?

Les tableaux de routage

- Permettent à établir des chemins entre n'importe quels deux utilisateurs
- Sont configurés dans des nœuds de lien : notamment les routeurs
- Le **routage statique** : programmer un tableau de routage à chaque nœud,
  - Écrit à la main par l'administrateur d'un réseau
  - Ne jamais le modifier
  - Peut être optimisé, mais fonctionne bien seulement dans des petits réseaux
- Le **routage dynamique** : les tableaux de routage s'adaptent dynamiquement
  - Les routeurs se parlent l'un à l'autre pour adapter leurs routes

Routage statique : solution 1

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	192.168.1.35
172.120.2.0/24	192.168.1.128



Programmer le table de routage

- Instruction :

  `ip route <add ou del> <destination> via <IP router>`

  - "Je veux que tout colis pour le réseau de destination xxx passe par le routeur R"
  - @Destination non nécessairement dans le même réseau que R
  - Un peu comme la conduite : la signalisation marche étape par étape

- Example : `ip route add 172.168.1.0/24 via 192.168.1.35`

- Pour voir les résultats :

  `ip route list`

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	192.168.1.35 
172.120.2.0/24	192.168.1.128 

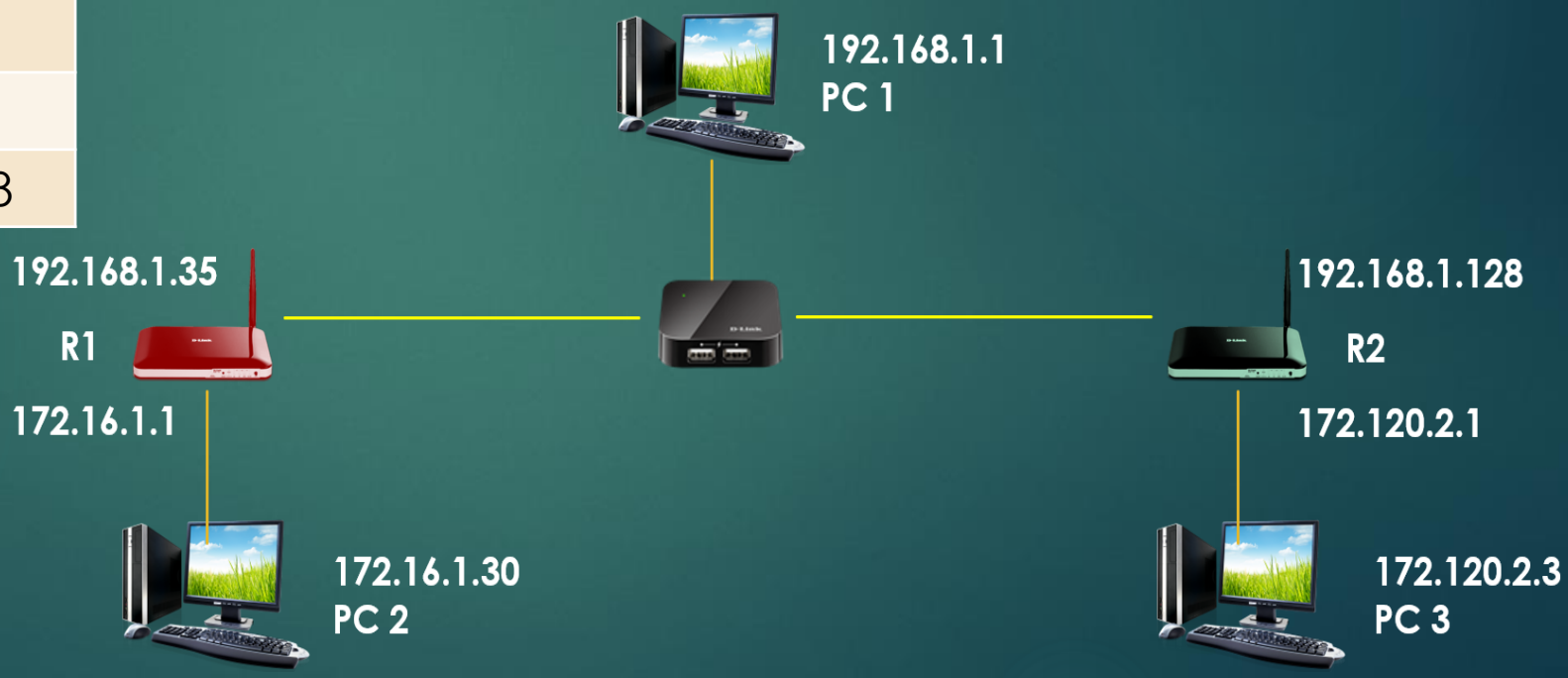
Une meilleure solution

- Notre solution précédente va dicter aux deux routeurs comment fonctionner
- Une meilleure solution serait pour PC 1 de choisir un router
  - Puis laisser le router faire ses choix par rapport aux destinations
  - Ceci facilite la programmation du routage pour les noeuds
  - … Et laisse la possibilité de juste changer/adapter le routage au niveau des éléments connecteurs -- les routers

Notamment...

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	none
172.120.2.0/24	192.168.1.128

Destination	Router
192.168.1.0/24	none
0.0.0.0/0	192.168.1.35



Implemmenter cette solution

- Pour PC 1 : un routage par défaut

```
ip route add default via 192.168.1.35
```

Destination	Router
192.168.1.0/24	none
0.0.0.0/0	192.168.1.35

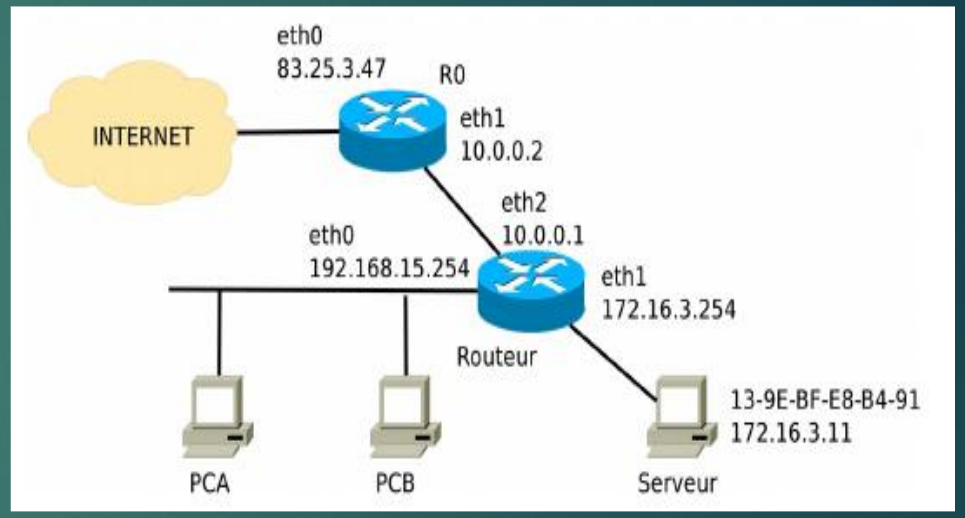
- Pour Routeur R1 :

```
ip route add 172.120.2.0/24 via 192.168.1.128
```

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	none
172.120.2.0/24	192.168.1.128

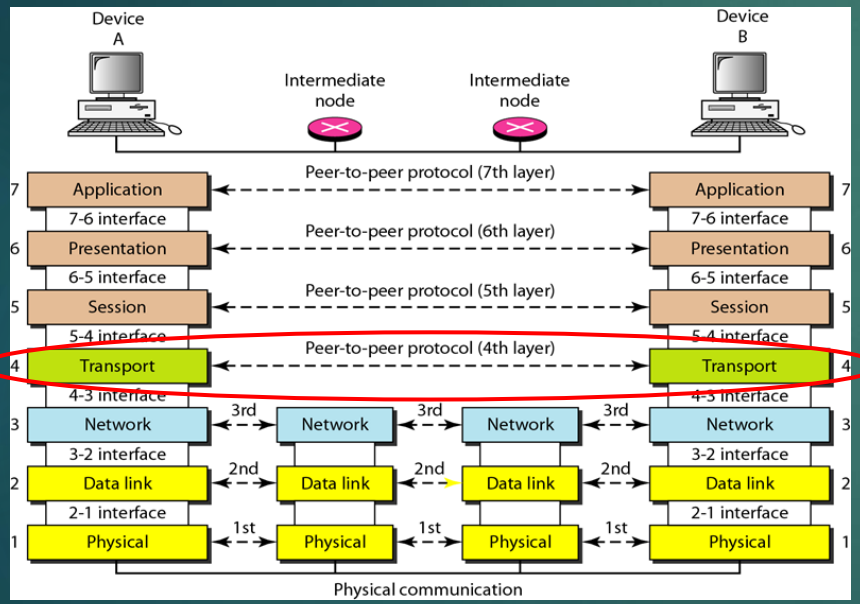
Route par défaut vs route spécifique

- Route par défaut :
  - Typique pour les machines des utilisateurs (PCA, PCB, Serveur)
  - Parfois utile pour indiquer une route "principale" pour un routeur (par ex. la route vers l'Internet) : R0 est la route par défaut de Routeur
- Route spécifique :
  - En général utilisée pour les routeurs, pour indiquer une route secondaire
    - Routeur est le routeur de R0 vers les deux sous-réseaux en bas



La couche transport

La couche transport



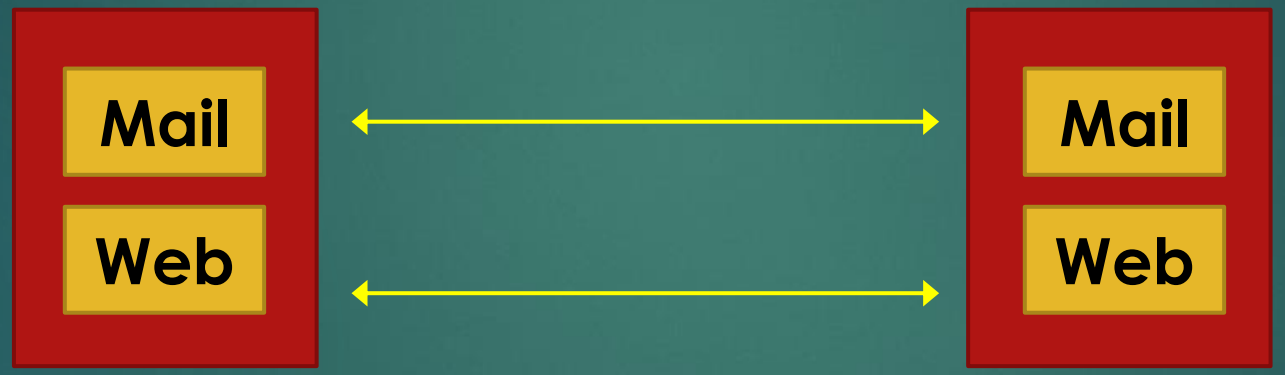
La couche transport

- **Objectif** : la fiabilité des communications
- Couche liaison : des trames d'information
  - Mais ces trames peuvent se perdre
  - Aucune notion de confirmation, fiabilité
- Couche transport : des sessions
  - Sessions de communication : flot bidirectionnel
  - Packets envoyés, reçu de reception
  - Rétransmission de packets non-reçus
  - Connexion coupée si transmission ne réussit pas

À cette couche : transmissions fiables, ports, protocoles UDP, TCP

La problématique des ports

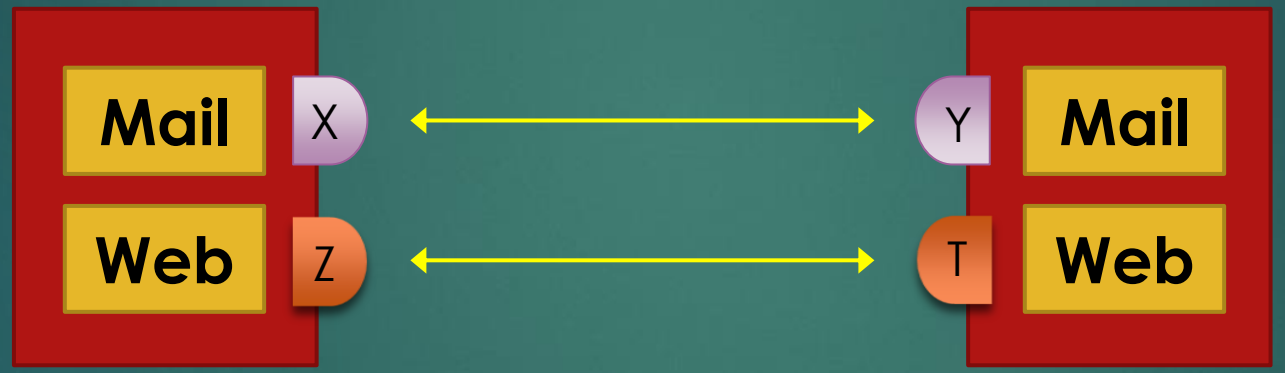
- Deux ordinateurs peuvent échanger des messages concernant plusieurs services



- Un serveur peut hôter plusierus services
- On veut que le bon message soit traité par le bon service

Solution : les ports

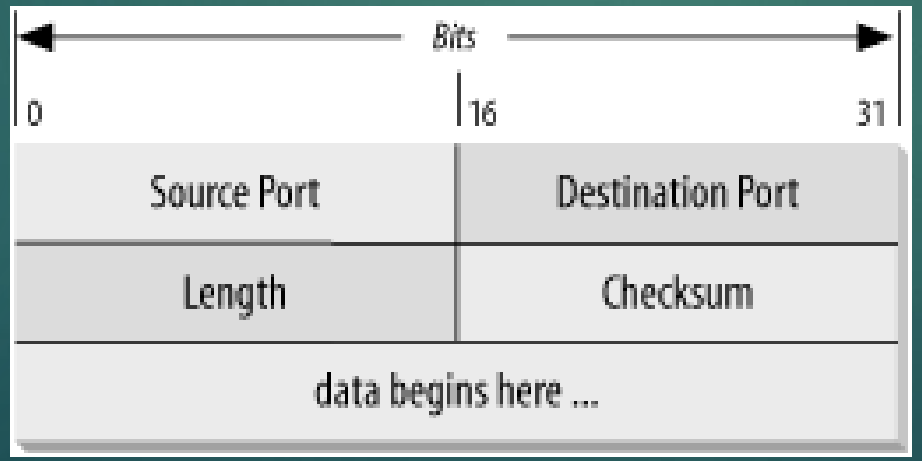
- Chaque service = un nouveau "port"



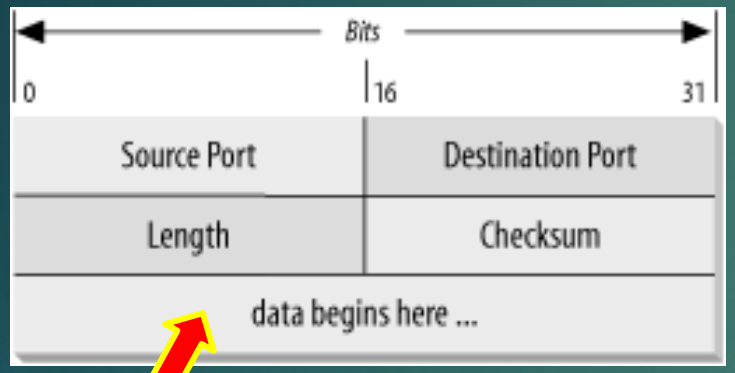
- 1 port = 16 bits
- Il y a des ports réservés
- 1 application ← → (adresse IP, Protocole, port)

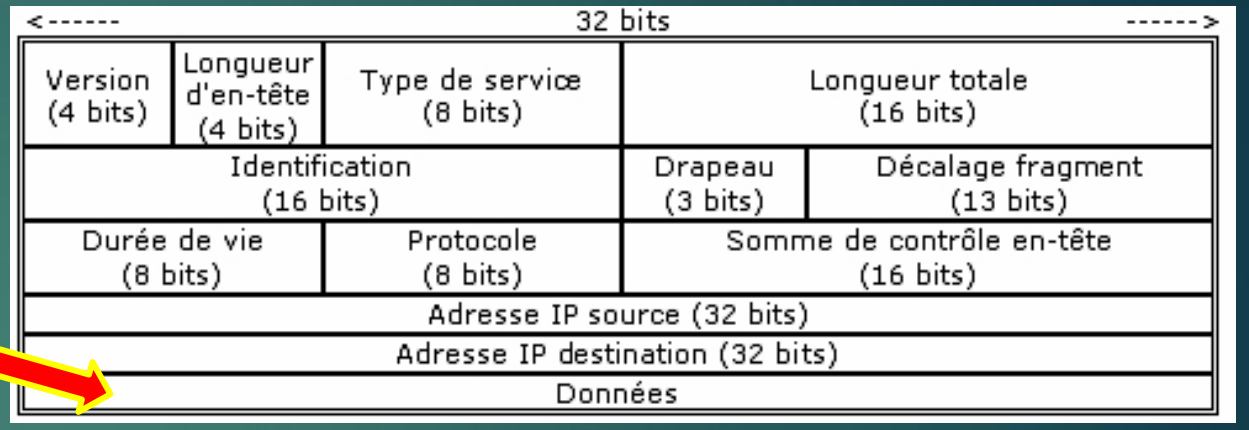
Le protocole UDP

- Un des protocoles le plus simples de la couche transport
- Messages = datagrammes (d'où UDP = User Datagram Protocol)
- Rajoute des informations sur les ports de destination et source
  - Très peu de fiabilité ajoutée en dehors des ports + checksum
  - Si on a besoin de plus de fiabilité, il faut utiliser TCP

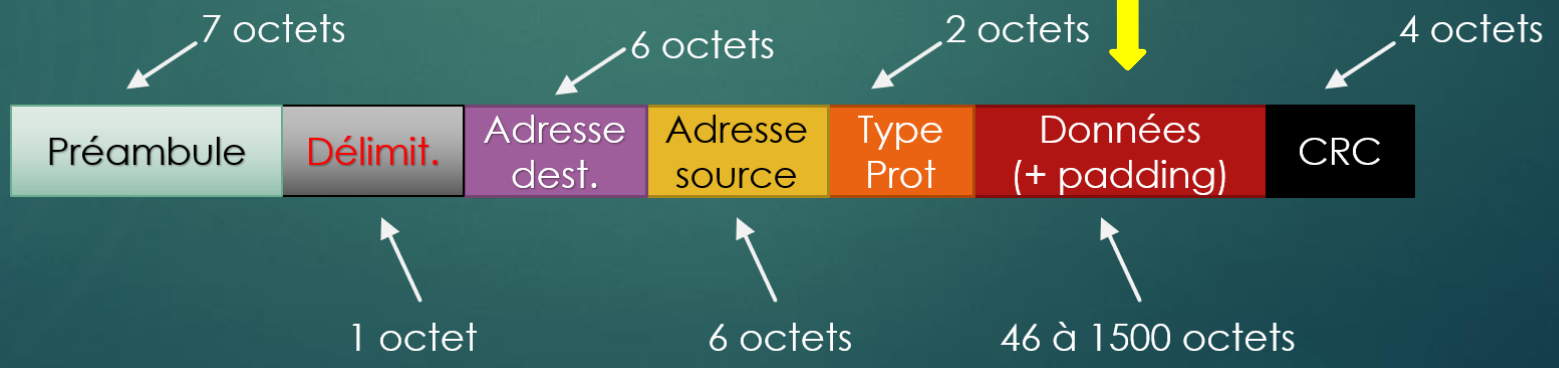


Encapsulation





Message

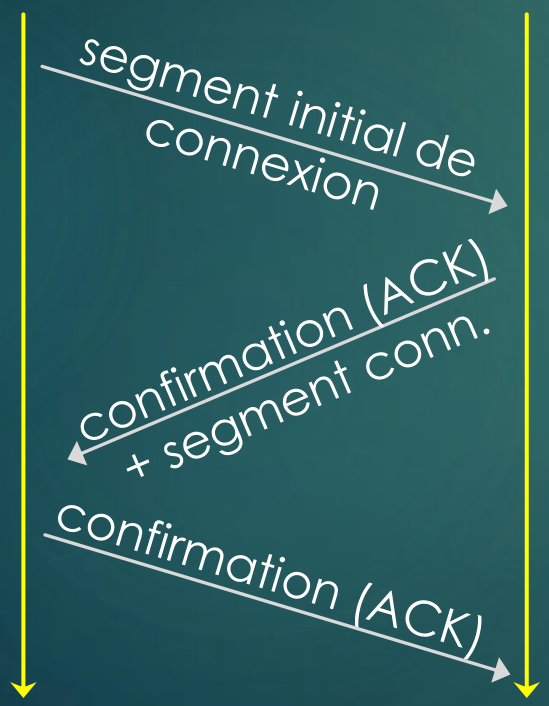


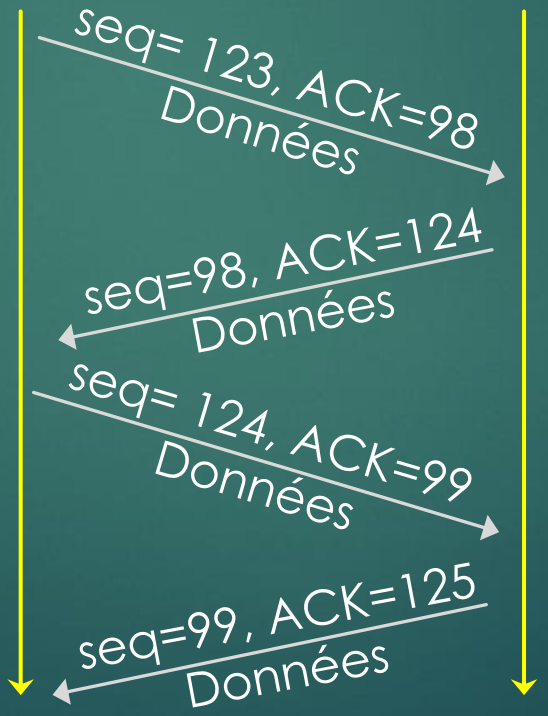
La transmission fiable : TCP

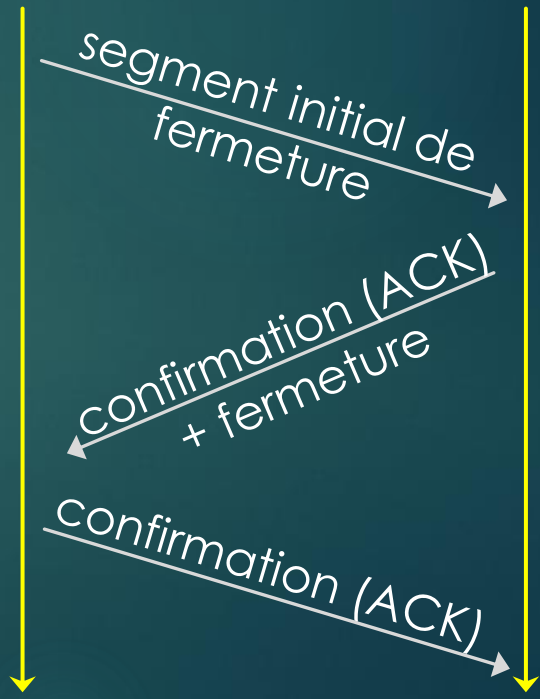
- Le protocole UDP permet :
  - La séparation de messages par services : la notion de port
  - Messages encapsulés dans le protocole IP, puis Ethernet
  - Pas de transmission fiable
- Le protocole TCP permet :
  - La transmission ordonnée d'un flux de messages
  - Correction d'erreurs
  - Transmission confirmée de messages

Connexion TCP

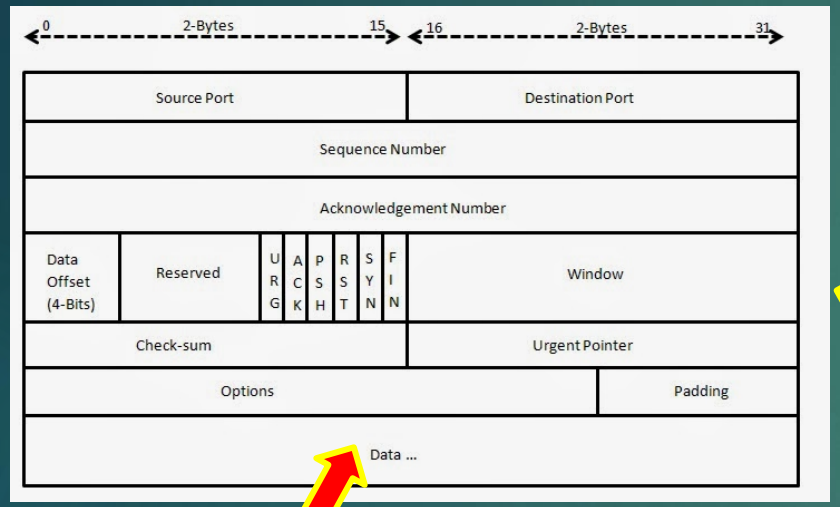
- La connexion s'ouvre avec une suite de messages
  - Puis les deux machine échangent des messages et confirment leur réception

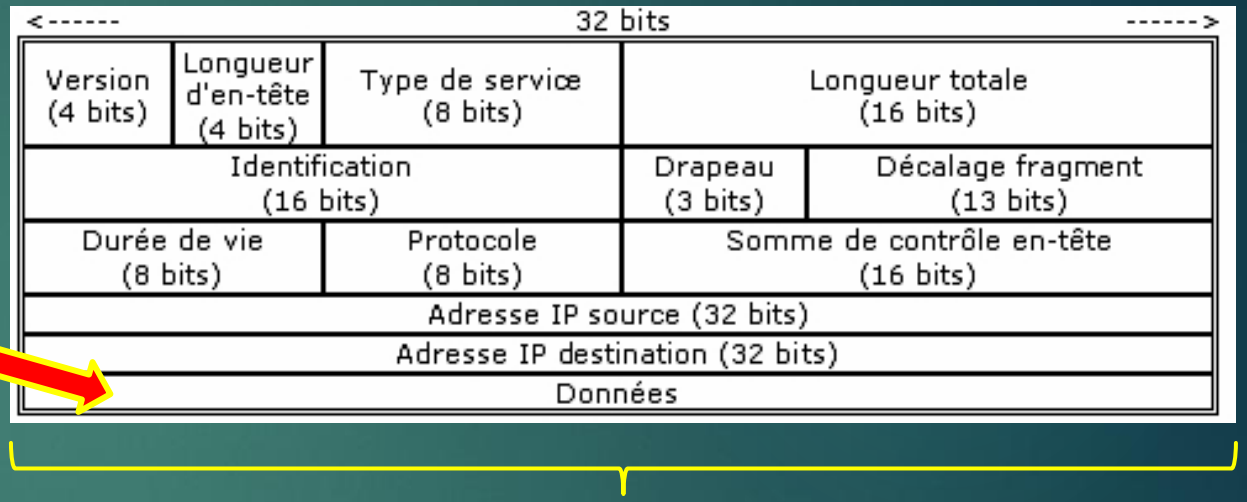




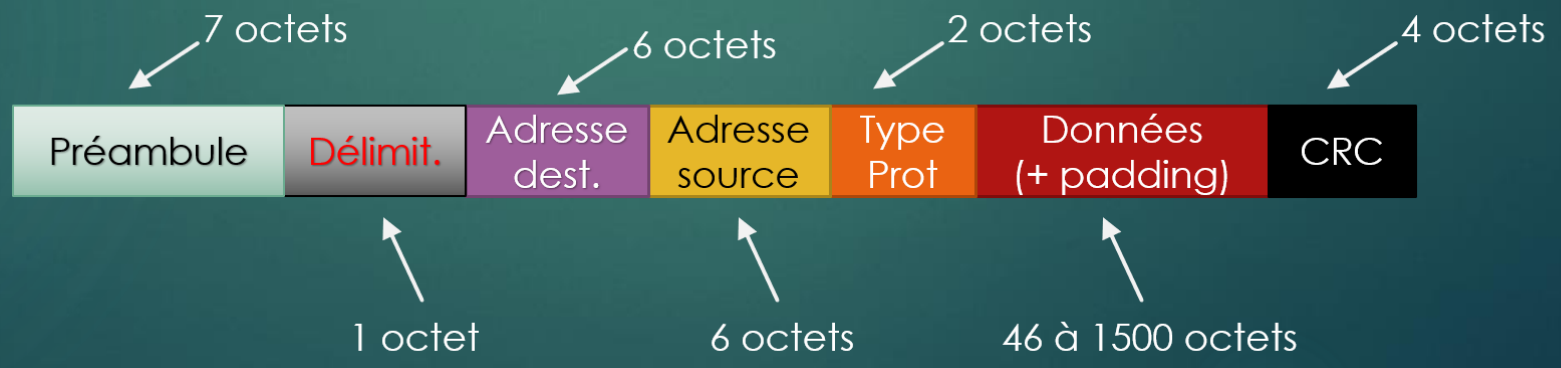


Encapsulation





Message



La commande ss

- Pour voir les connexions actuelles d'une machine : la commande ss
  - Connexions liées à des protocoles différents (applications différentes)
  - Statut : à l'écoute, en cours, en cours de finir...

- Permet de visualizer toute connexion en cours
  - Paramétrisation possible :
    - ss -ln ports écoute (en format numérique)
    - ss -uan toute connexion sur udp, format numérique
    - ss -tn connexions sur TCP, format numérique