

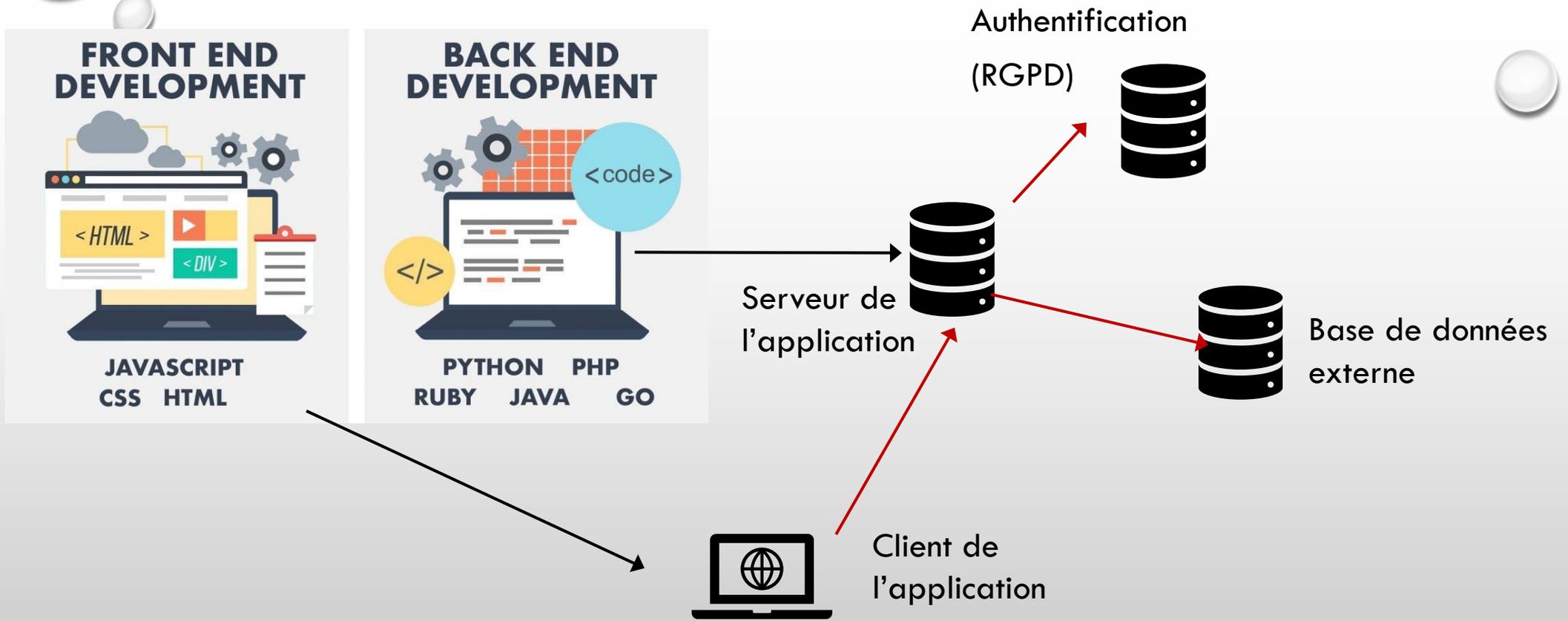


# R2.04 – LES BASES DES RÉSEAUX

**RESPONSABLE : CRISTINA ONETE**

**MATERIEL : [HTTPS://WWW.ONETE.NET/TEACHING.HTML](https://www.onete.net/teaching.html)**

**EMAIL : [MARIA-CRISTINA.ONETE@UNILIM.FR](mailto:MARIA-CRISTINA.ONETE@UNILIM.FR)**



# DÉVELOPPEMENT ET RÉSEAUX

# EXEMPLE : GESTIONNAIRE DES MISSIONS

Développement :  
Cahier de charges

Appli WEB  
suivi missions

3 rôles : chef,  
employé,  
compta

Employé :  
renseigne trajet,  
trouve billets

Chef : OK/KO  
mission, charges

Compta : suivi  
financier, retour

Chaque étape :  
notif. par mail

Déposer  
justificatifs

Développement :  
Concept → Application

- Analyse de besoins
- Consultations client
- Conception + conception tests
- Maquettage
- Développement (versioning)
- Tests, Adaptation
- Consultations client

# EXEMPLE : GESTIONNAIRE DES MISSIONS

Développement :  
Cahier de charges

Appli WEB  
suivi missions

3 rôles : chef,  
employé,  
compta

Employé :  
renseigne trajet,  
trouve billets

Chef : OK/KO  
mission, charges

Compta : suivi  
financier, retour

Chaque étape :  
notif. par mail

Déposer  
justificatifs

Intégration, Réseaux

- Quel type de serveur, où l'héberger ?
- Accès intranet ou internet ?
- Comment gérer la messagerie ?
- Cartographie échange messages : quel service, quel port, d'où vers où ?
- Accès bases de données ?
- Maintenance : manuelle/CDN/cloud...

# OBJECTIFS : R2.04 + R2.05

## OBJECTIF 1

### MAÎTRISER LES BASES DU FONCTIONNEMENT D'UN RÉSEAU INFORMATIQUE

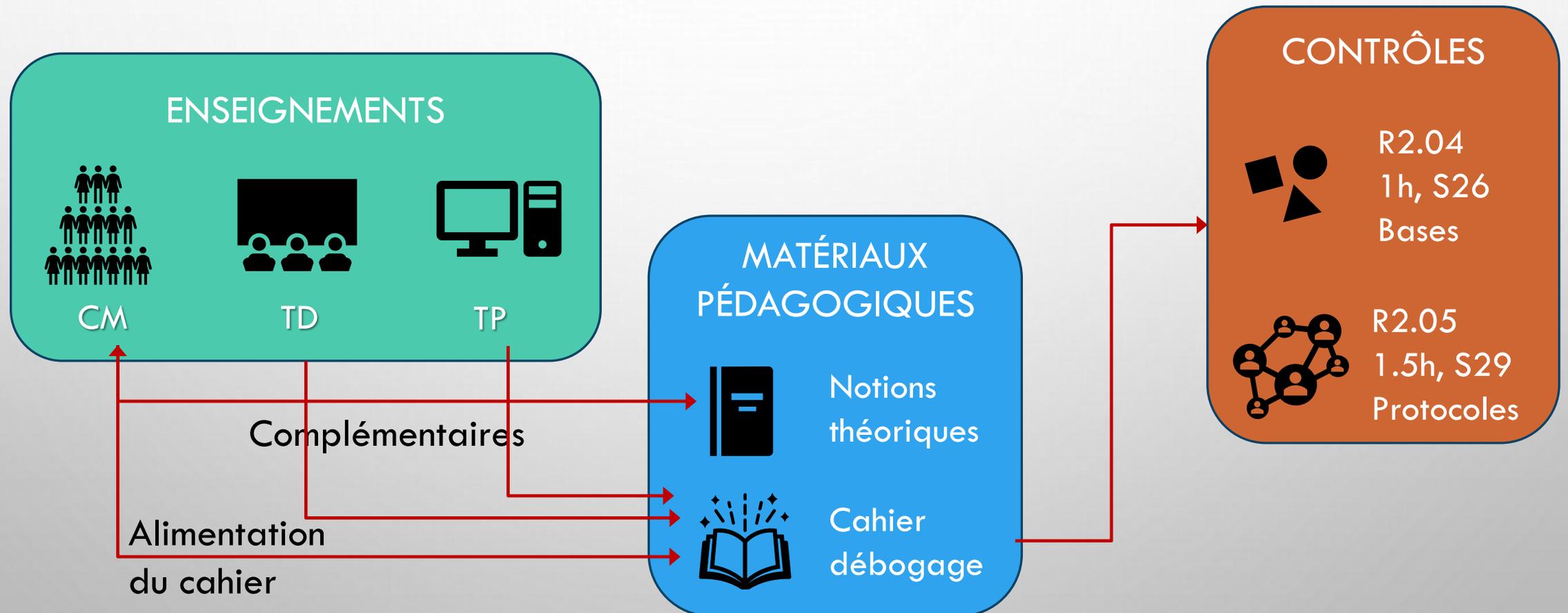
- Modèle OSI, modèle TCP/IP
- Adresses IP, adresses MAC
- Routage statique, transmission intra- et inter-réseaux
- Débogage et résolution d'un problème de connectivité

## OBJECTIF 2

### UTILISER DES PROTOCOLES RÉSEAUX

- Protocoles client-serveur
- Configuration et utilisation DHCP
- Configuration et utilisation d'un serveur HTTP ou FTP
- Configuration client DNS, résolution directe et inverse

# NOTIONS THÉORIQUES, DÉBOGAGE, CONTRÔLES

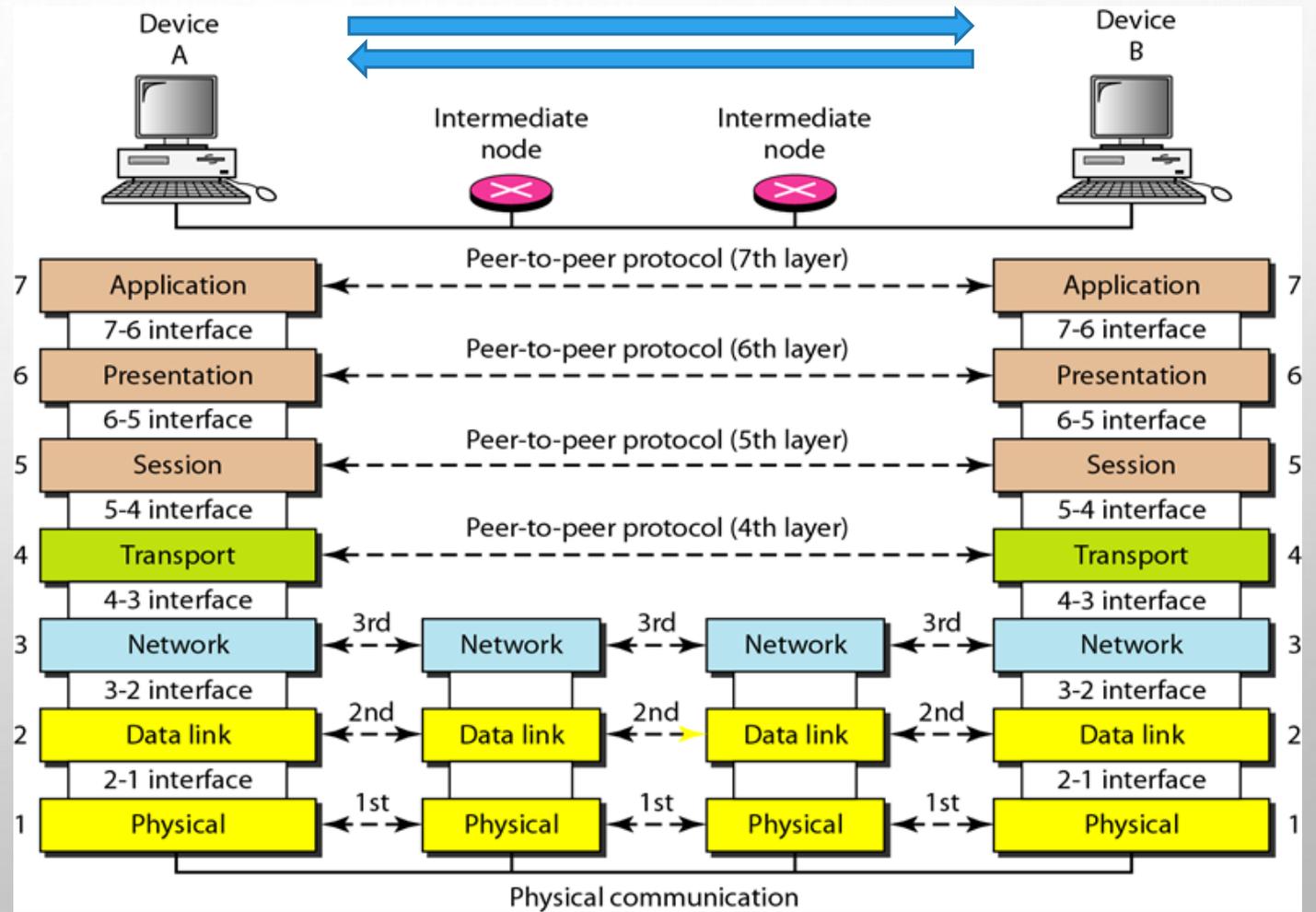


The background features a light blue gradient on the left side, transitioning to a light gray gradient on the right. On the left, there are several water droplets of various sizes, some overlapping. Below the droplets, a network structure is visible, consisting of white nodes connected by thin white lines, resembling a mesh or a web. The overall aesthetic is clean and modern.

# LA STRUCTURE DES RÉSEAUX

LES MODELES OSI ET TCP/IP

# COMMUNICATION SUR RÉSEAU



# LES COUCHES OSI



Open Systems Interconnection



Modèle d'abstraction/  
communication en réseau :

Universel à tout type de réseau



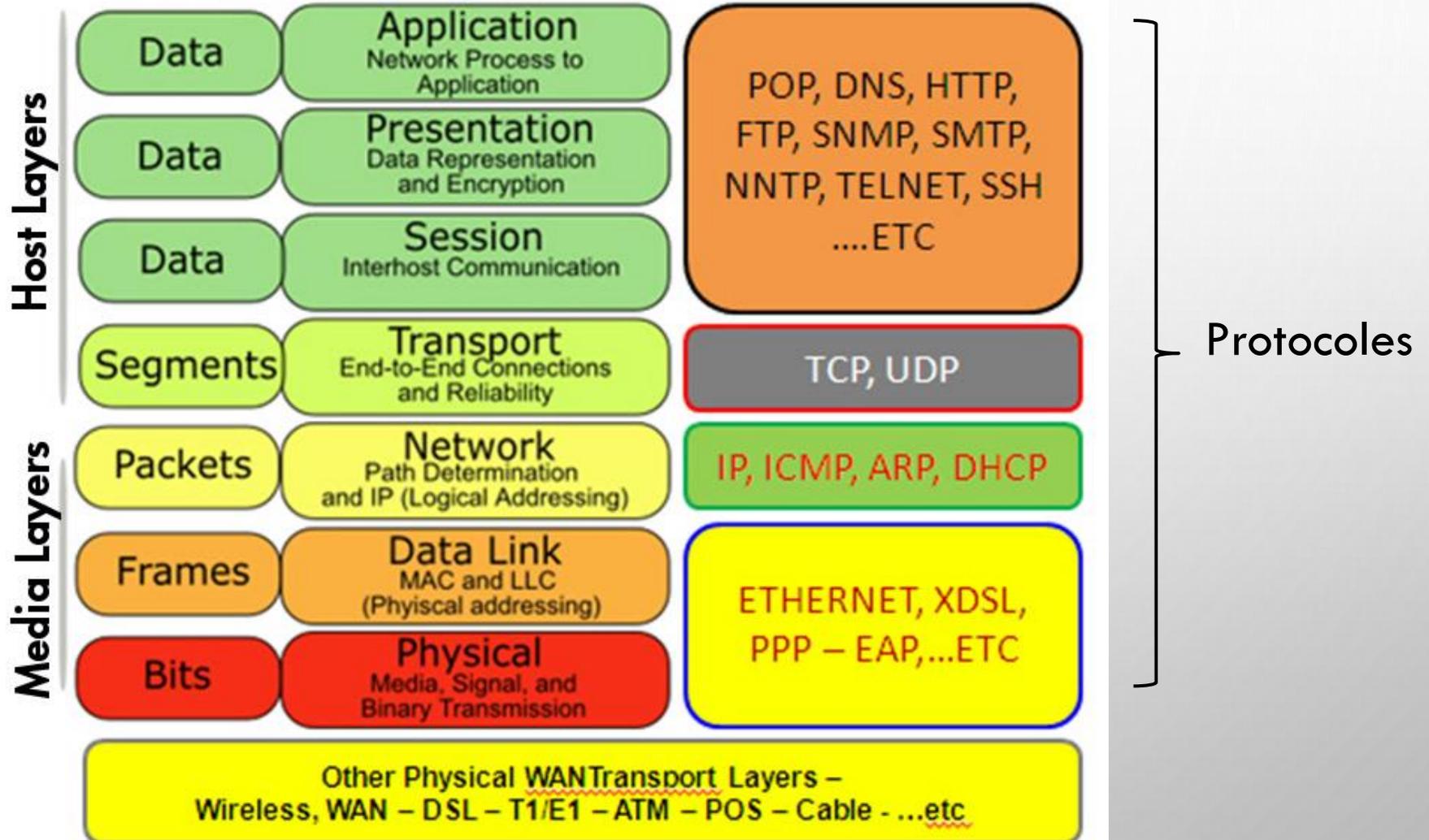
7 couches indépendantes :

Couches basses – vers le medium physique

Couches hautes – vers l'application

Indépendantes, à role bien défini

### OSI Example for Ethernet Media - TCP/IP STACK

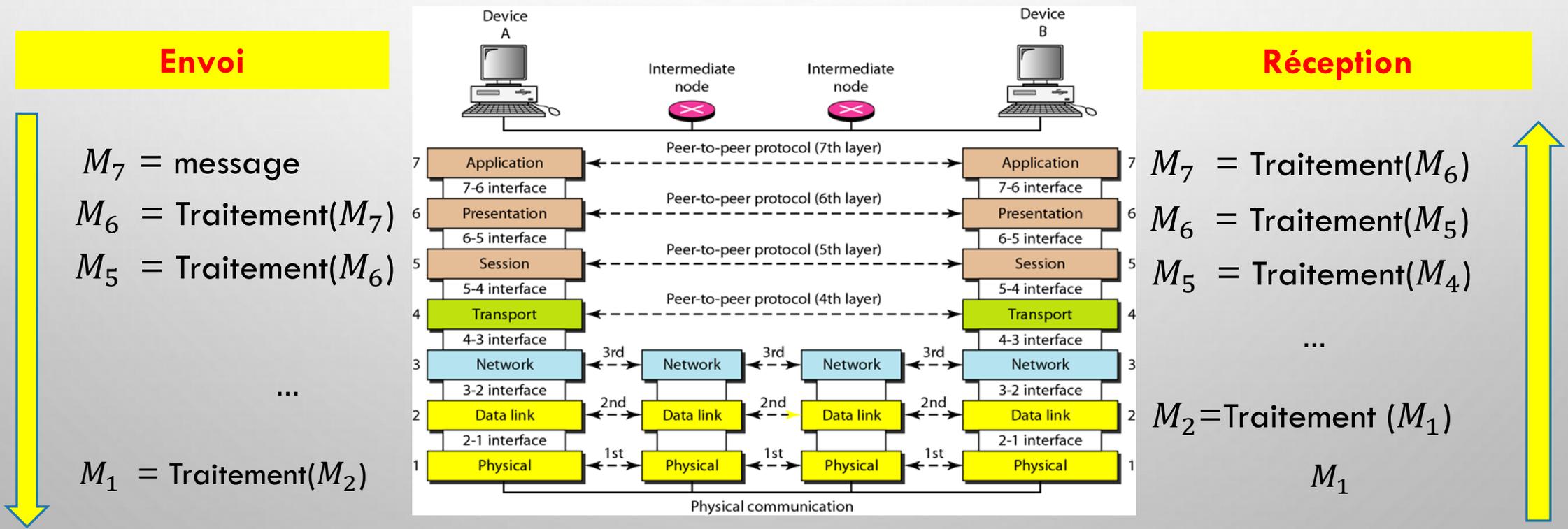


Couches architecturales

Protocoles

# ENCAPSULATION ET DÉCAPSULATION

- TRAITEMENT DES MESSAGES À L'ENVOI/RECEPTION : COUCHE PAR COUCHE
  - RAJOUT DE PRÉFIXES, SUFFIXES POUR AIDER LE TRAITEMENT



# L'INDÉPENDANCE DES COUCHES

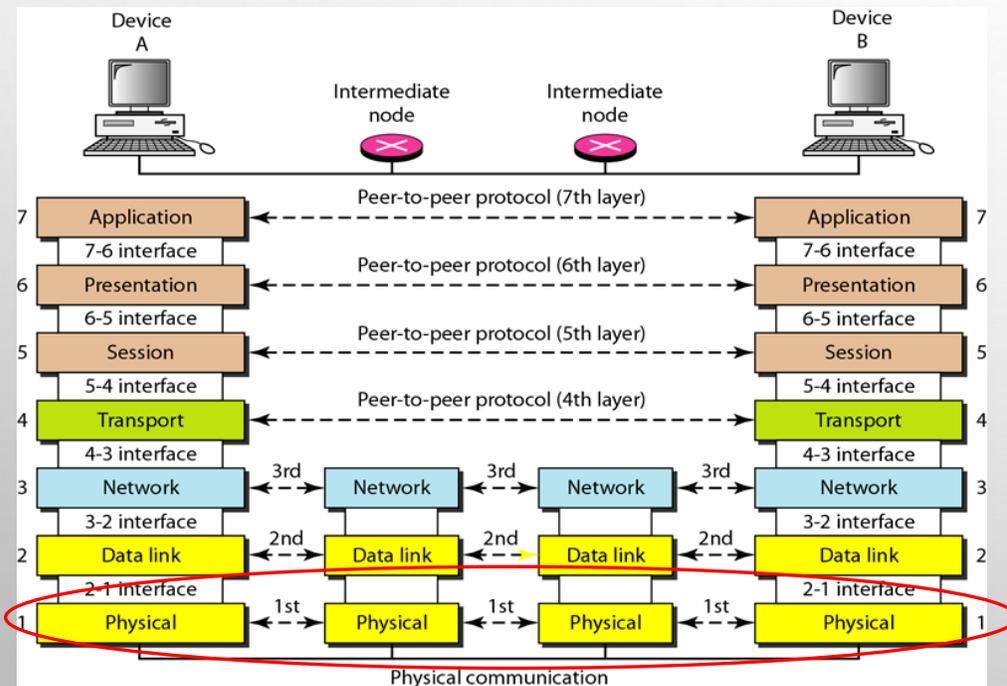
- CHAQUE COUCHE RAJOUTE UNE PROPRIÉTÉ
  - Peu importe le contenu du message
  - Peu importe pourquoi le message a été envoyé
- COUCHE 3 : ENVOIS ENTRE DES RÉSEAUX DIFFÉRENTS
  - ... pour un message d'erreur
  - ... pour une résolution de nom
  - ... pour une requête HTTPS...
- ENCAPSULATION COUCHE 3 → DÉCAPSULATION COUCHE 3



# LA COUCHE PHYSIQUE



# DESCRIPTION DE LA COUCHE PHYSIQUE



- TOUTE COMMUNICATION PASSE PAR UN MEDIUM PHYSIQUE
- COMMUNICATION POSSIBLE SEULEMENT SI UNE CONNEXION PHYSIQUE EXISTE
- DIFFÉRENTS TYPES :
  - CABLE (CUIVRE) : COURANT = 1; AUCUN COURANT = 0
  - FIBRE : LUMIÈRE = 1; PAS DE LUMIÈRE = 0
  - DES ONDES RADIO (CONNEXION SANS FIL)

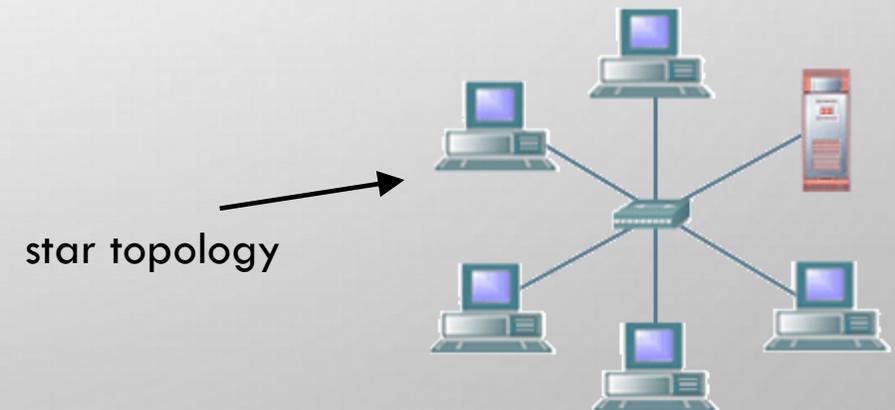
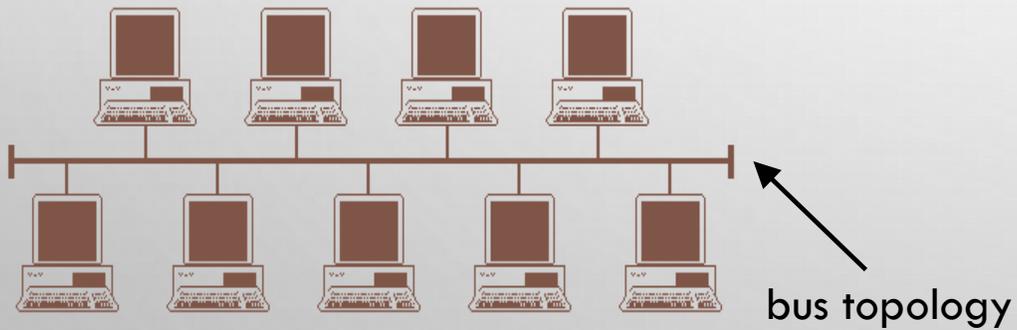
**Nous ferons très peu à cette couche ici**

# ELÉMENTS DE TOPOLOGIE

- CABLES TYPIQUES : CUIVRE, FIBRE, ...
- CONNECTEURS : HUB/ SWITCH



- TOPOLOGIES TYPIQUES :



## HUB VS. SWITCH

Hub :  
broadcast

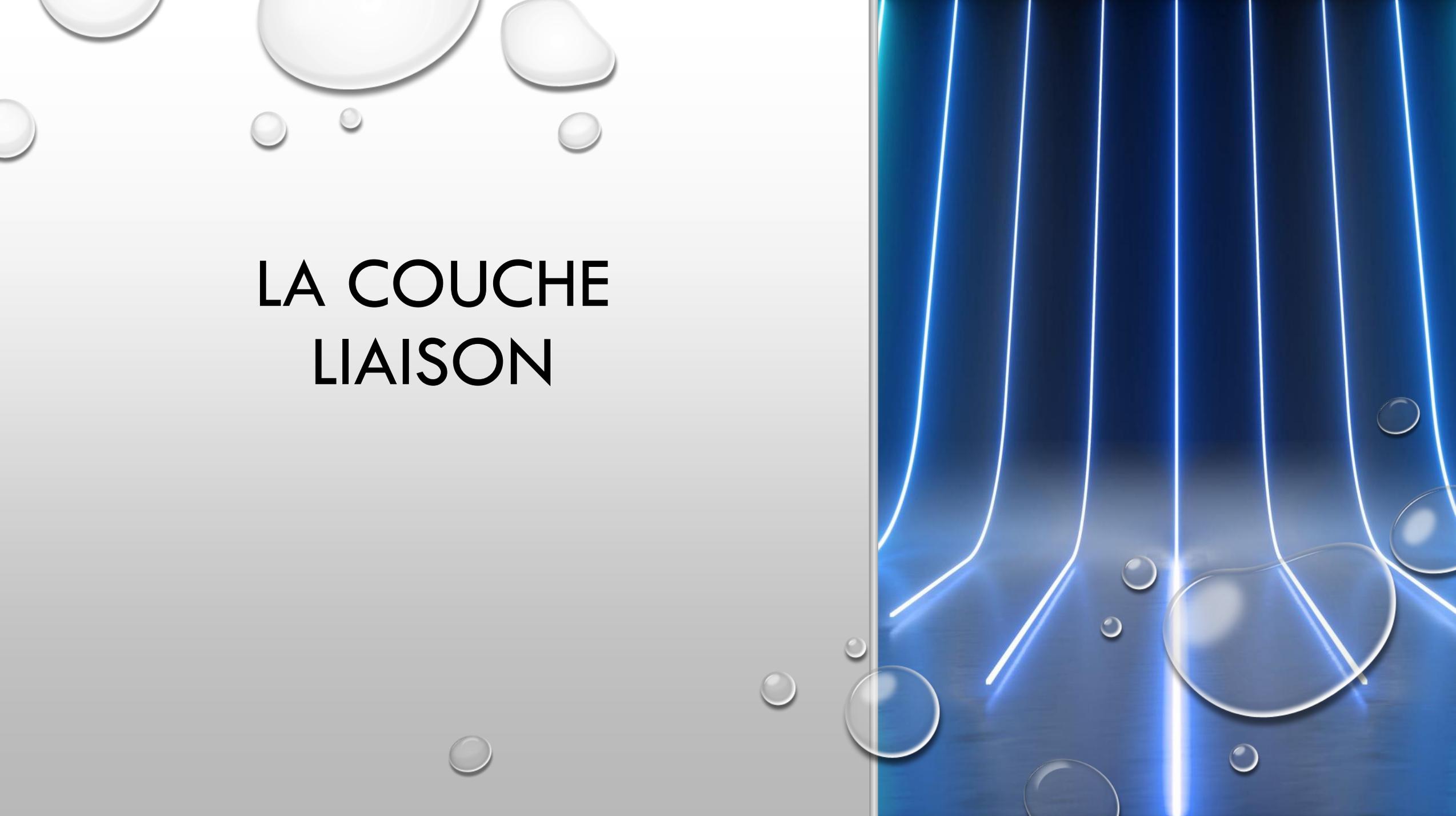


- Message arrive à toute machine connectée
- Très utile aux broadcasts
- Overhead inutile si unicast

Switch :  
unicast



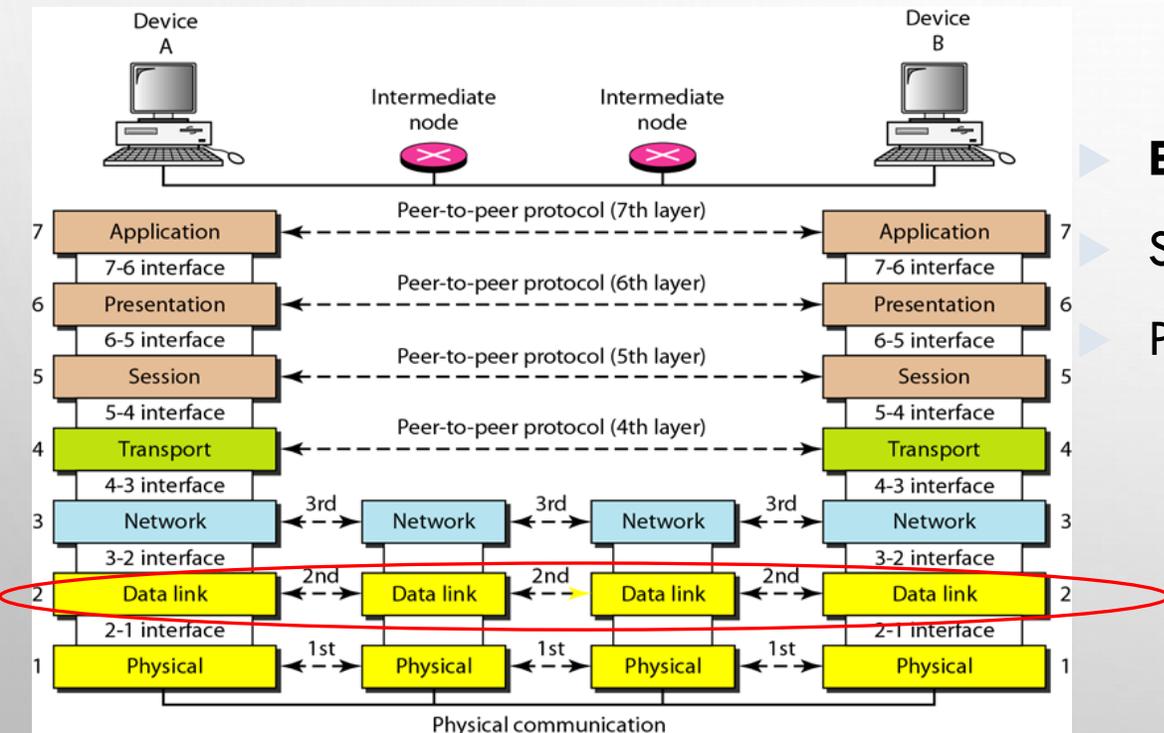
- Message envoyé à des destinataires précis
- Utile aux unicasts
- Inefficace pour les broadcasts



**LA COUCHE  
LIAISON**

# LA COUCHE LIAISON

## TRANSMISSION INTRA-RÉSEAU (MÉDIUM PHYSIQUE PARTAGÉ)



▶ Envoi de messages intra-réseau simultanés

▶ Signaux (bits) → trames

▶ Protocoles de transmission simultanée comme Ethernet

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

# LES ADRESSES MAC

- COUCHE LIAISON : COMMENT ADRESSER DES MESSAGES ?
- LES ADRESSES MAC :
  - 48 bits = 6 octets en deux parties
  - uniques et permanentes
- ADRESSES SPÉCIALES : FF:FF:FF:FF:FF:FF = BROADCAST, 00:00:00:00:00:00 = INCONNUE

33: 33 : XX : XX : XX : XX POUR MULTICAST

3 octets Network Interface Controller

00 – 50 – 56 – C0 – C6 – 01

3 octets Organisationally Unique Identifier  
identifie le fournisseur

# TROUVER LES ADRESSES MAC

- INTERFACES RÉSEAUX : IDENTIFIANTS NUMÉRIQUES  $\leftrightarrow$  CARTES RÉSEAUX

- Chaque interface aura sa propre adresse MAC (et IP)

- EN WINDOWS : IPCONFIG

- COMMANDES À UTILISER (NON-ROOT) :



- ip address show : configurations toute interface
- ip address show up : config interfaces actives

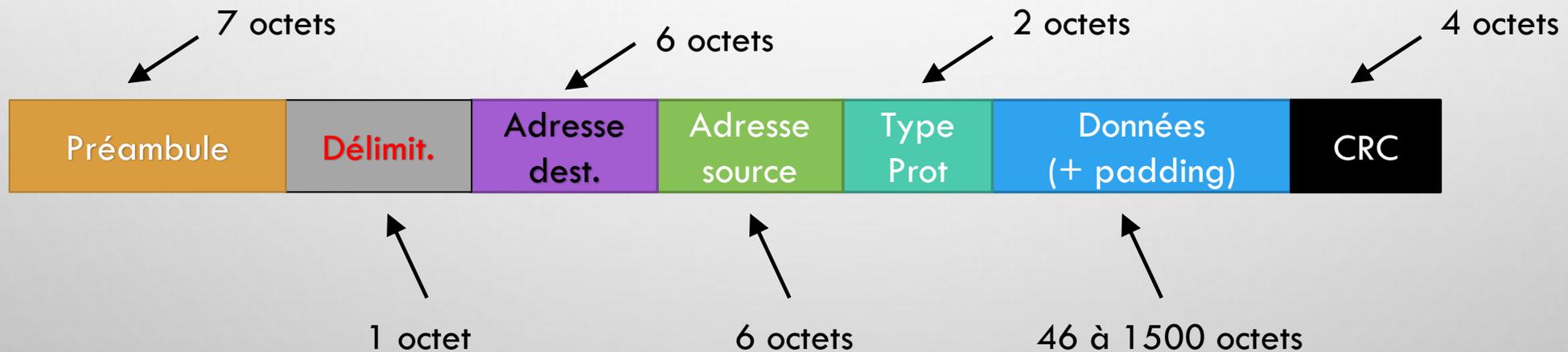
interface eth0

eth0

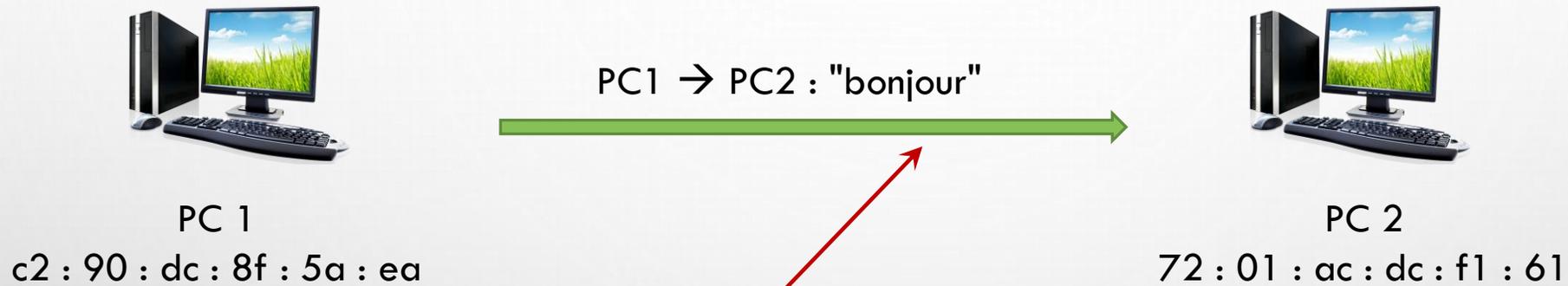
```
Link encap:Ethernet HWaddr 9e:6a:95:21:5e:c0
inet addr:192.168.10.2 Bcast:192.168.10.255 Mask:255.255.255.0
inet6 addr: fe80::9c6a:95ff:fe21:5ec0/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:4987 errors:0 dropped:0 overruns:0 frame:0
TX packets:3108 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:7331287 (6.9 MiB) TX bytes:215819 (210.7 KiB)
Interrupt:5
```

## LA TRAME ETHERNET II

- ENTRE 46 ET 1500 OCTETS
  - DU PADDING (BOURRAGE) SI TAILLE INFÉRIEURE



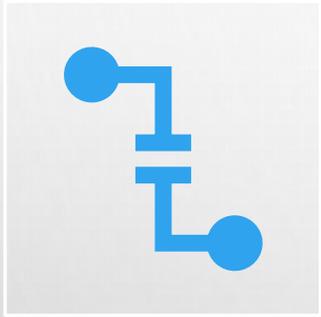
# HELLO WORLD : COUCHE LIAISON



```

▶ 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: 626f6e6a6f75720101010101010101010101010101010101...
  [Length: 34]
  
```

# PROTOCOLES COUCHE 2

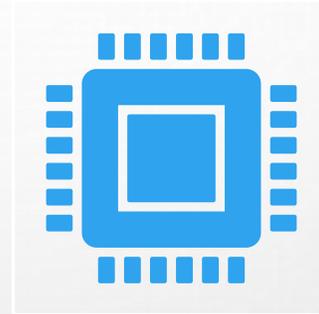


## Protocole “de couche 2”

Encapsulation couche 1 : OUI

Encapsulation couches 3 et + : **NON**

Fonctionne **intra-réseau**



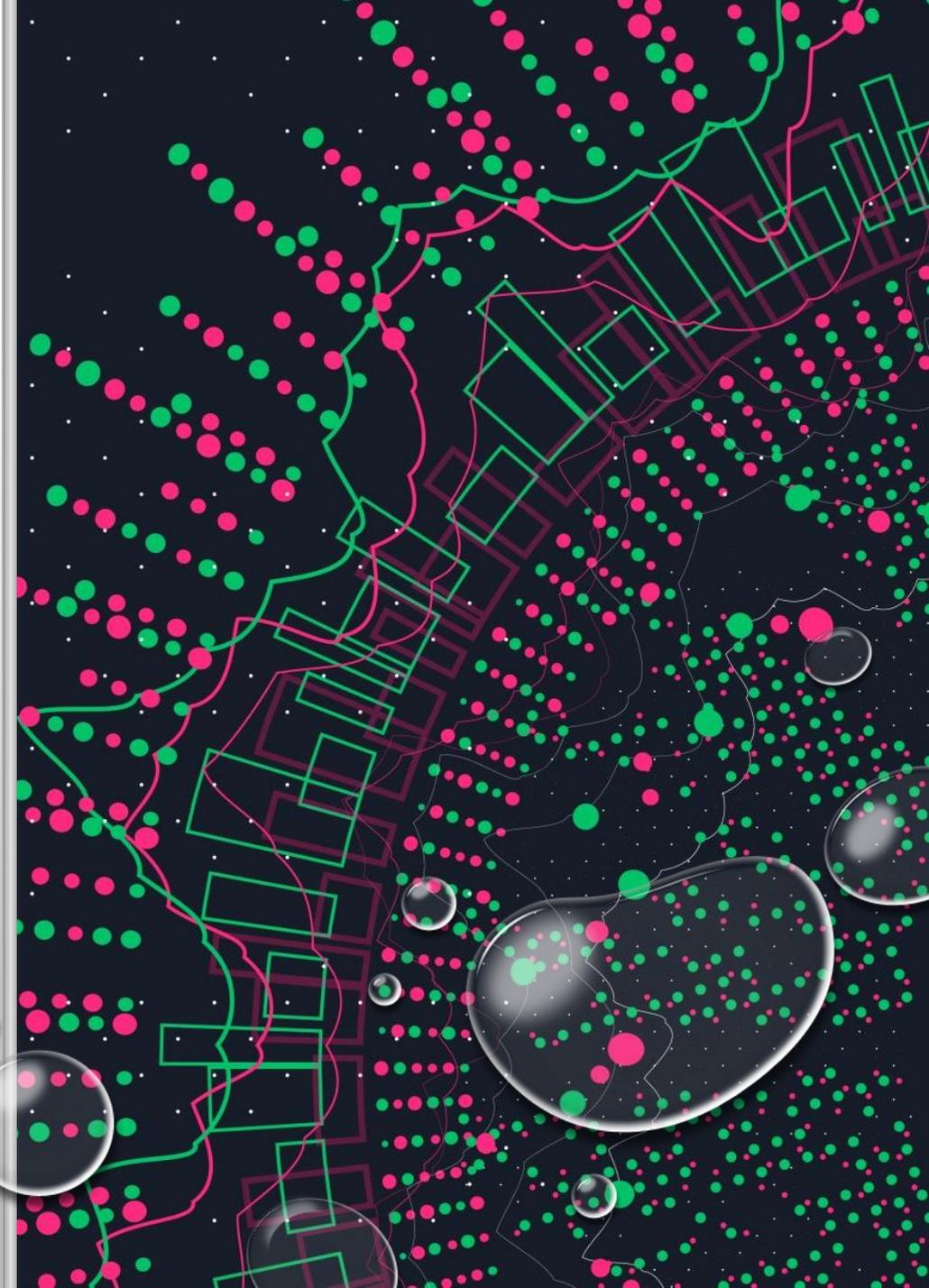
## Encapsulation couche 2

Encapsulation couche 1 : OUI

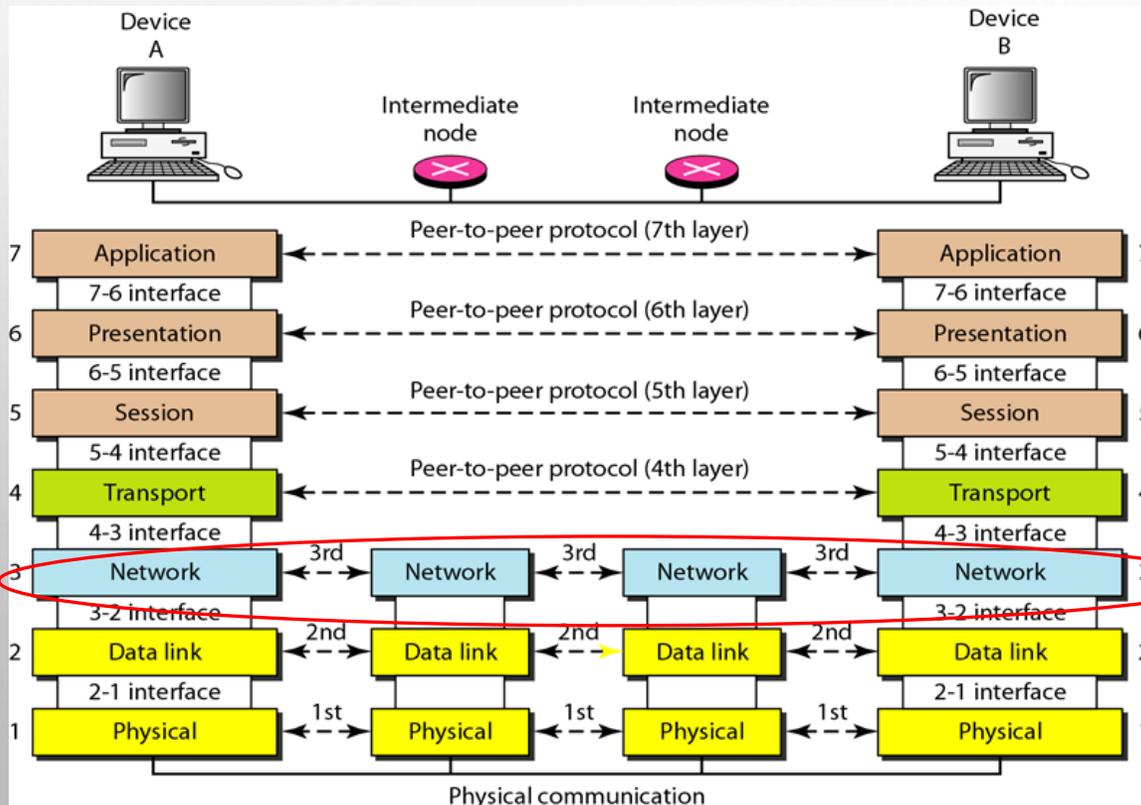
Encapsulation couche 3 et + : **OUI**

Fonctionne **inter-réseau**

# LA COUCHE RÉSEAU



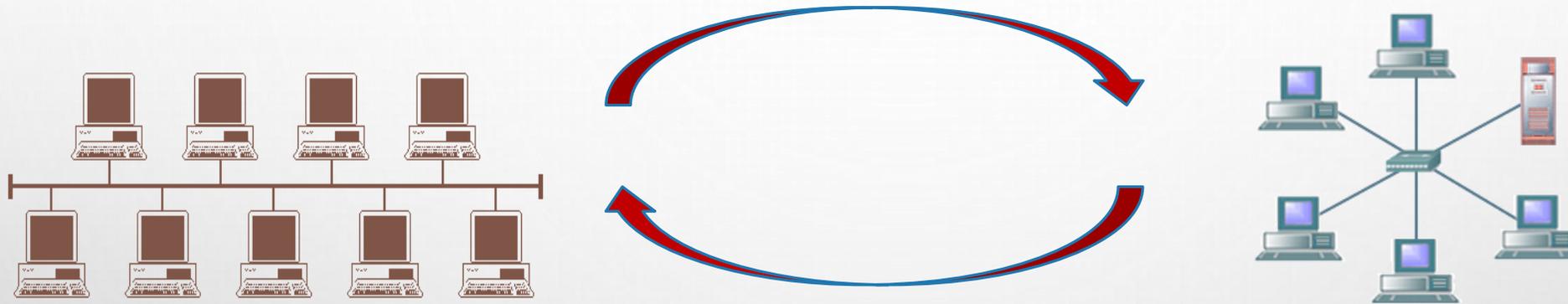
# LA COUCHE RÉSEAU



- ▶ 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 COMMUNICATION INTER-RÉSEAUX

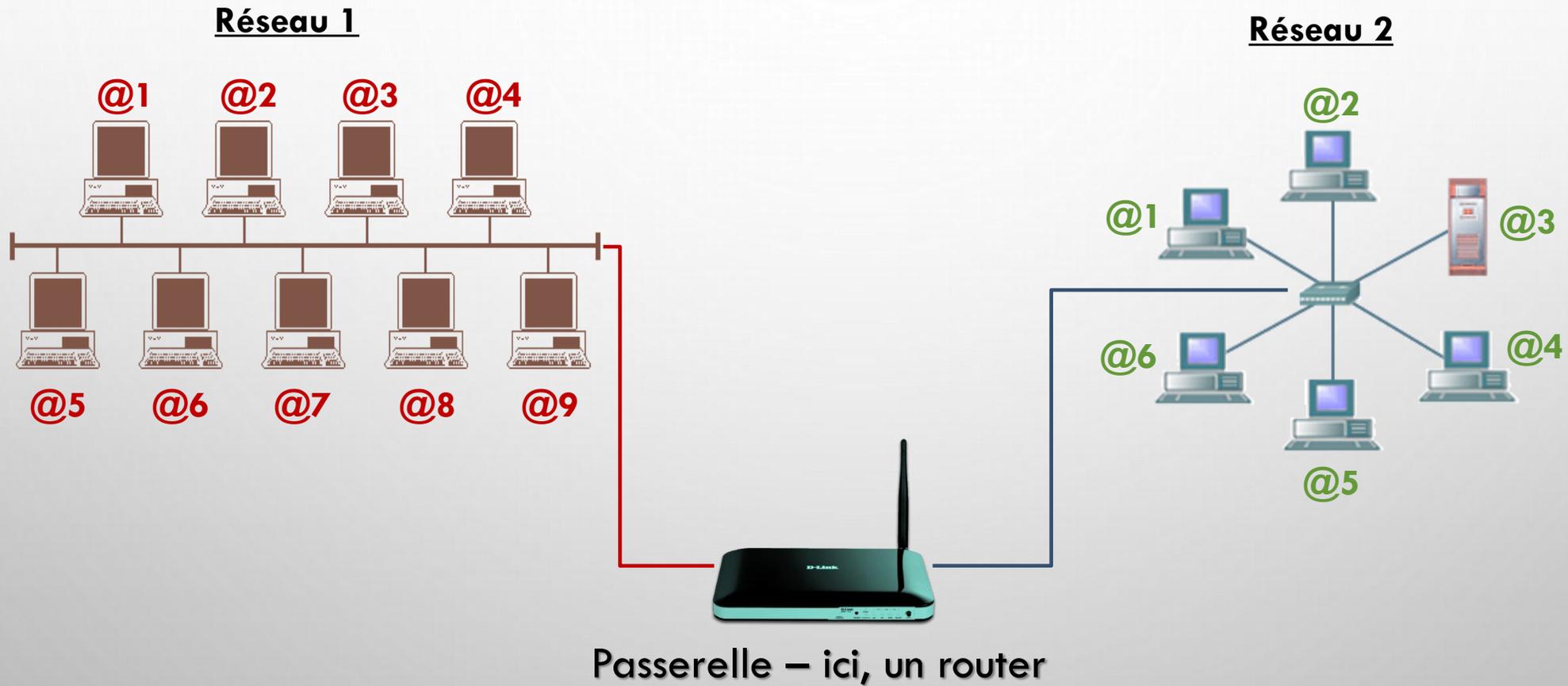


Comment envoyer des messages entre les deux réseaux ?

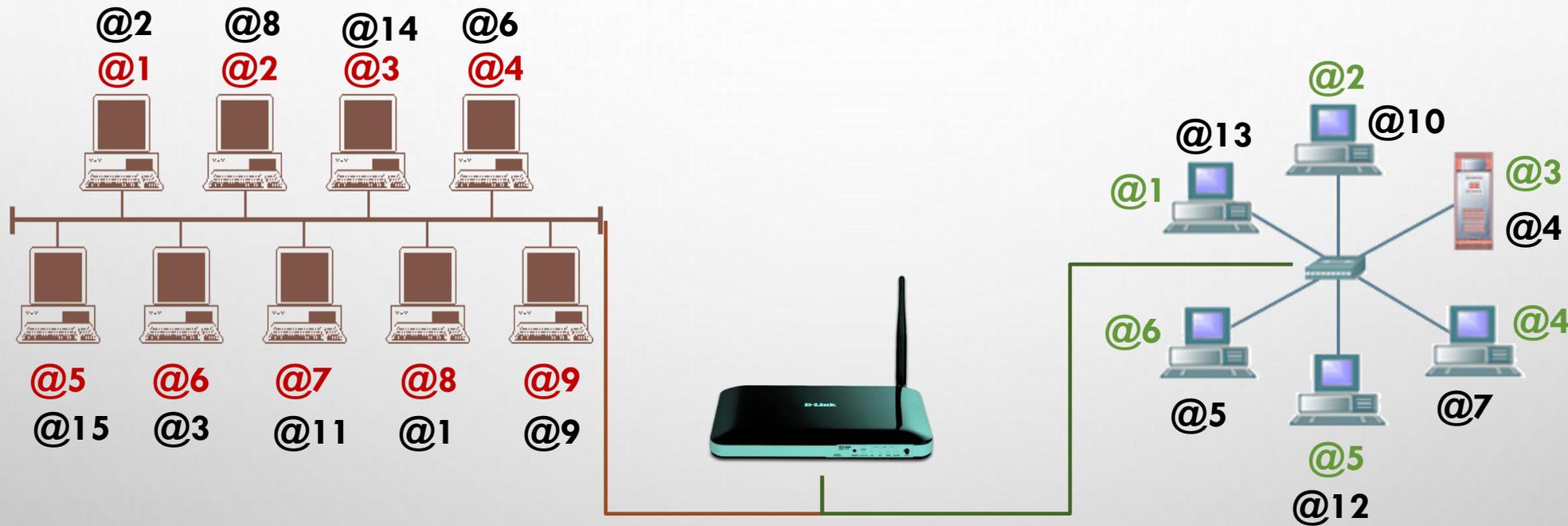
Communication couche 2

Communication couche 2

# LA PASSERELLE



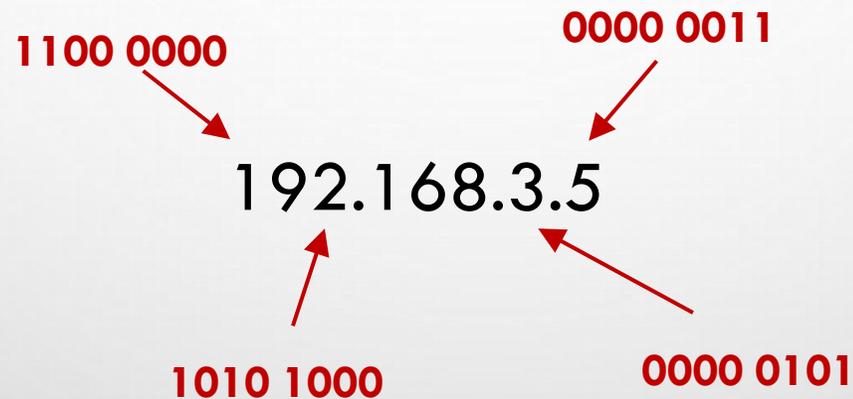
# ADRESSE MAC, ADRESSE IP



- ADRESSE MAC : IDENTIFIE LE PÉRIPHÉRIQUE
  - UNE MACHINE PEUT BOUGER, PEUT CHANGER DE RÉSEAU
- UNE ADRESSE “LOGIQUE” QUI IDENTIFIE LE RÉSEAU DE CHAQUE MACHINE : **LES ADRESSES IP**

# ADRESSE IP, DOMAINE DE COLLISION

- EN IPv4 : 32 BITS = 4 OCTETS SÉPARÉS PAR DES POINTS



- L'ADRESSE IP EST **TEMPORAIRE** ET **NON-UNIQUE**
  - ❖ PRÉFIXE INDIQUE LE RÉSEAU DUQUEL LA MACHINE FAIT PARTIE
  - ❖ RÉSEAU COMMUN = DOMAINE DE COLLISION

# DOMAINE DE COLLISION, CIDR

- DOMAINE DE COLLISION = suite d'adresses, souvent contigües, même préfixe
  - Préfixe commun = un nombre de bits (sur 32), constant sur toutes les adresses
- TAILLE DU PRÉFIXE => TAILLE DU RÉSEAU
  - ❖ PRÉFIXE À X BITS (SUR 32) => TAILLE DU RÉSEAU SERA DE  $2^{32-x} - 2$ 
    - Une adresse IP = 32 bits.
    - x bits constants (préfixe) => 32-x bits pour les adresses machine
    - Chaque bit est 0 ou 1, d'où  $2^{32-x}$  possibilités machine.
    - 2 adresses prises : réseau, broadcast
- TAILLE D'UN DOMAINE DE COLLISION => NOTATION CIDR : 192.168.5.23/24
  - ❖ /24 = préfixe 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
- **/24 => MASQUE DE** 11111111.11111111.11111111.00000000 = 255.255.255.0
- PARTIE RÉSEAU : LES 24 PREMIERS BITS : 192.168.5
  - ❖ **ADRESSE DU RÉSEAU** : préfixe, suivi par des 0s et CIDR : 192.168.5.0/24
- PARTIE MACHINE :
  - ❖ DERNIÈRE ADRESSE : **@ BROADCAST**
    - ❖ Préfixe, suivi par 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

# CONFIGURER UNE ARCHITECTURE

- **PROBLÉMATIQUE** : besoin d'une sous-plage pour 75 machines.

Quelle taille de sous-réseau ?

- ❖ CIDR sur 26 bits ?  $2^6 - 2 = 64$  machines
- ❖ CIDR sur 25 bits :  $2^7 - 2 = 126$  machines
- ❖ Configuration : typ...

Comment configurer une adresse IP ?

- **DÉCOUPAGE** plage /X en deux plages /X+1

- ❖ Identifier le préfixe de la plage initiale
- ❖ Découper la plage en 2 parties disjointes (x+1<sup>ème</sup> bit) :
  - ❖ Sous-réseau 1 : préfixe initial suivi par 0
  - ❖ Sous-réseau 2 : préfixe initial suivi par 1

12.23.128.0/18

12.23.[10000000].0

12.23.128.0/19

12.23.[10000000].0

12.23.160.0/19

12.23.[10100000].0

# CONFIGURER UNE MACHINE

- **PROBLÉMATIQUE** : Configurer l'adresse 192.168.3.129 dans le réseau 192.168.3.128/25
- ÉLÉMENTS À CONFIGURER : @IP, masque CIDR, l'interface à configurer
  - ❖ MÉTHODE 1 : Configuration temporaire, utiliser la commande ip

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

```
ip link set dev eth0 up
```
  - ❖ MÉTHODE 2 : configuration pérenne, modifier le fichier /etc/network/interfaces  
ON VERRA CETTE MÉTHODE PLUS TARD

Configurer une adresse ne la rend pas immédiatement visible !

# ADRESSE VALIDE, INVALIDE

- COMMENT VÉRIFIER QU'UNE ADRESSE EST CORRECTE ?

- ❖ Une adresse IPv4 n'a que 4 octets

- ❖ Chaque octet prend une valeur entre 0 et 255

**23.15.289.1 : invalide**

- ❖ Adresse machine vs réseau : une adresse réseau ne contient que des 0s après le préfixe

**122.12.65.0/23 :  
valide ou invalide ?**



**122.12.[0100 0001].0/23**

# LES CLASSES DE RÉSEAUX

- 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 la 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

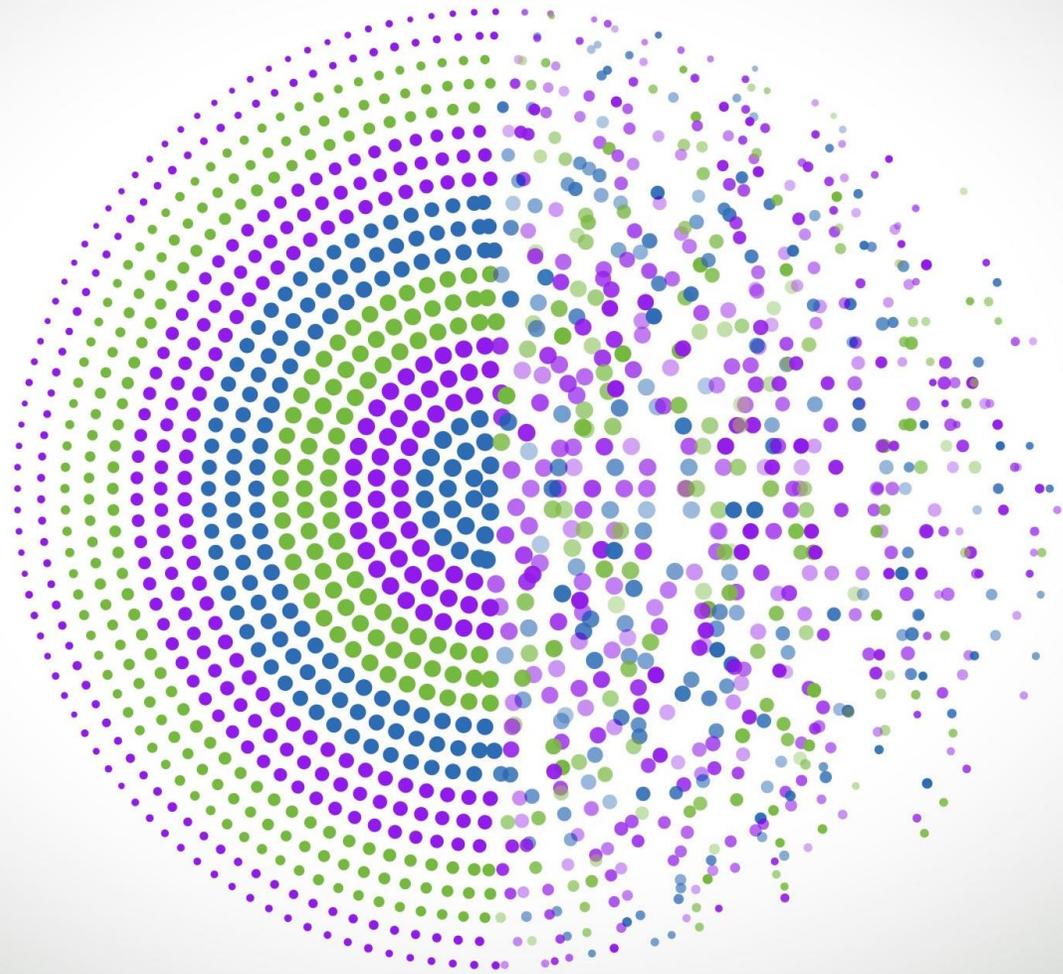
# ADRESSES SPÉCIALES

Adresse de broadcast : par défaut la dernière de la plage

Adresse de loopback : 127.0.0.1

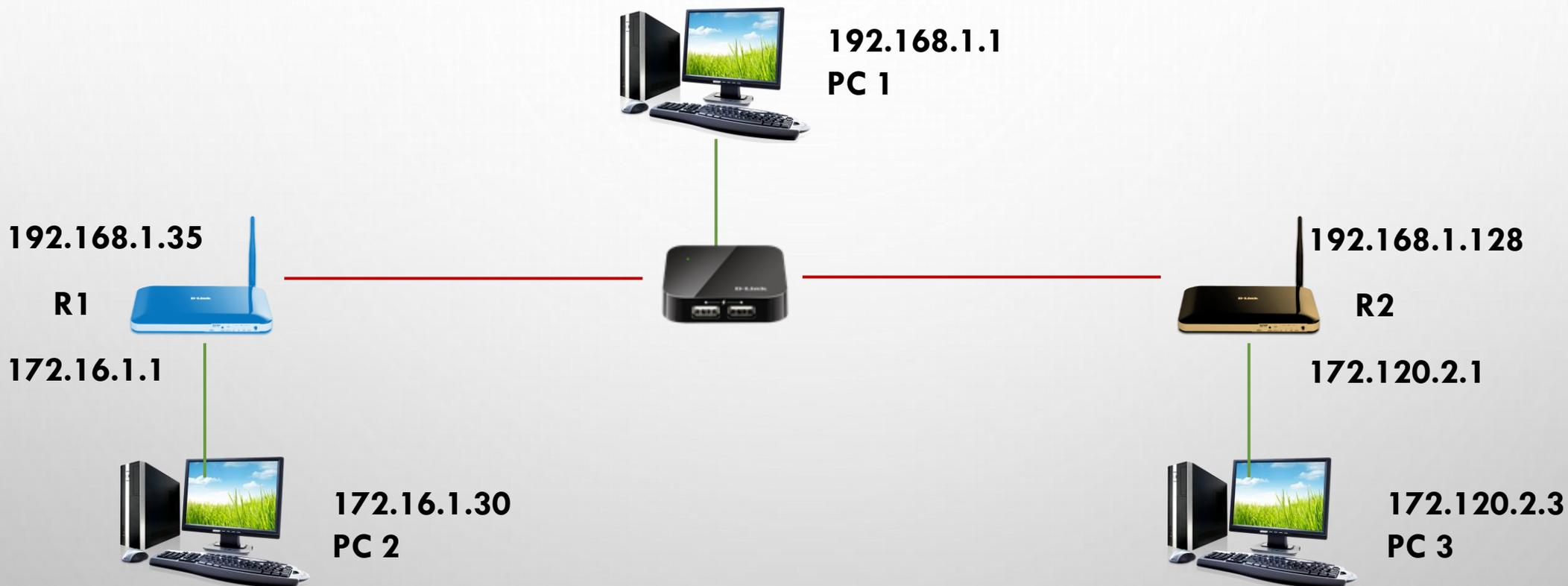
Adresses privées :

- 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**



# 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

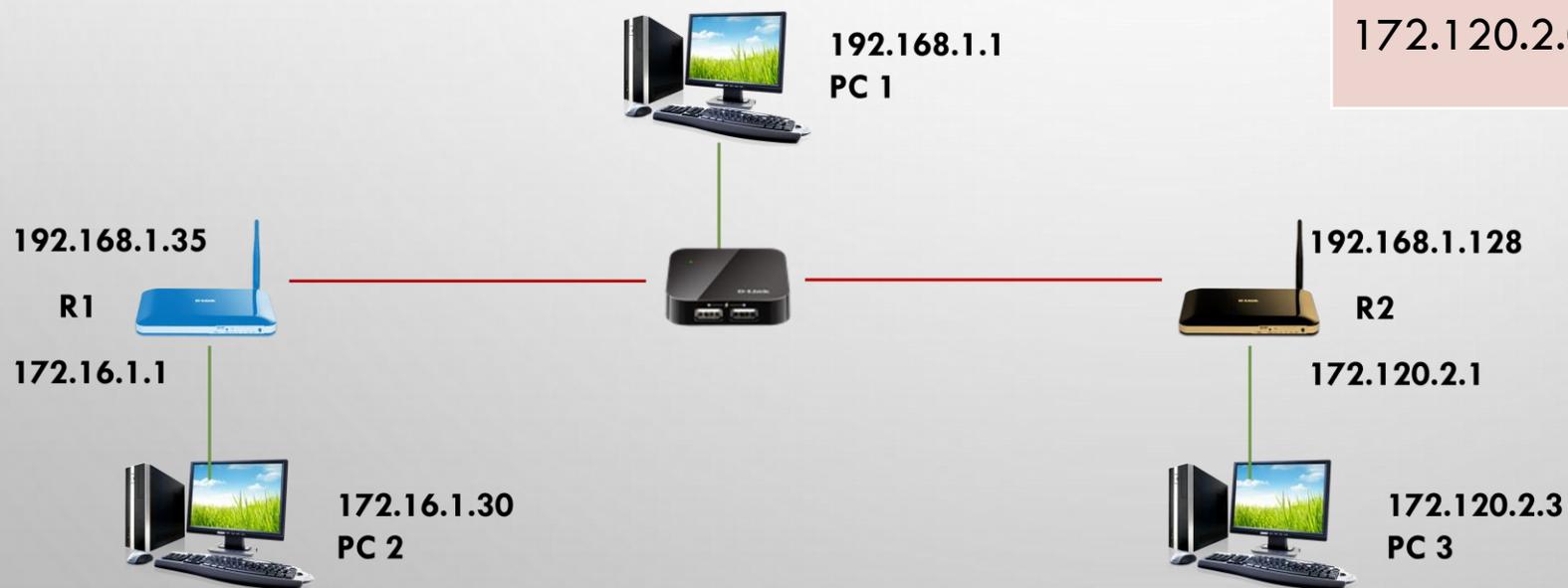
- **OBJECTIF** : Établir des chemins entre n'importe quels deux utilisateurs
- Configurés dans des nœuds de lien (routeurs)
- **ROUTAGE STATIQUE** : tableau de routage programmé à chaque nœud,
  - Écrit à la main par l'administrateur de réseau à chaque modification
  - Peut être optimisé, fonctionne bien seulement dans des petits réseaux
- **ROUTAGE DYNAMIQUE** : tableaux de routage qui s'adaptent dynamiquement
  - Les routeurs se parlent l'un à l'autre pour adapter leurs routes

# LE ROUTAGE : UNE AUTRE VUE

- CIRCULATION : A CHAQUE POINT, UNE DIRECTION VERS LE PROCHAIN POINT
  - Au rondpoint 1 : prenez à droite
  - Plus tard : prenez à gauche
  - Au rondpoint 2 : tout droit
- ROUTAGE : CHAQUE ROUTEUR, PROCHAINE ÉTAPE
  - R1 : prochain routeur : R2
  - R2 : pour aller vers 192.168.1.0/24, R3  
pour tout autre chemin, R0
- ON N'INDIQUE JAMAIS TOUT LE CHEMIN



# 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

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

**@dest. finale des packets**

Routage toujours nécessaire hors réseau, jamais dans le réseau

- EXEMPLE: `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

Solution précédente : OK mais non-optimale



Une meilleure solution serait de :

**Choisir routeur  
par défaut PCA**

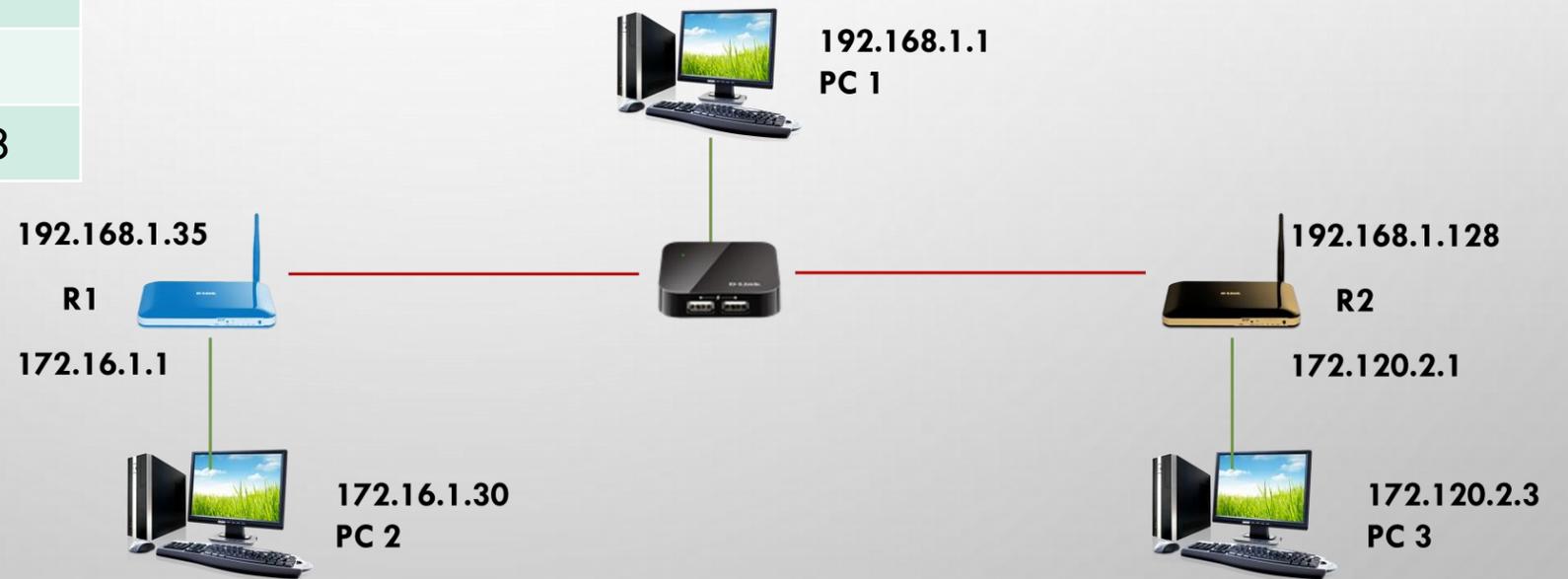
**Ce routeur décide  
le routage plus  
loin**

**PLUS FACILE :**  
routage plus facile  
à modifier

**PLUS FLEXIBLE :**  
routage  
modifiable selon  
besoins et routeurs

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
```



- POUR ROUTEUR R1 :

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



Destination	Router
192.168.1.0/24	none
0.0.0.0/0	192.168.1.35

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 machines utilisateurs (PCA, PCB, Serveur)
  - Peut indiquer une route "principale" pour un routeur (ex. la route vers l'Internet)
- ROUTE SPÉCIFIQUE :
  - En général utilisée pour les routeurs, pour indiquer une route secondaire

