

– R2.04 – Communication et fonctionnement bas niveau –

Cours 1: Introduction aux réseaux

Minh Tan PHAM

BUT INFO, 2022-2023

IUT de Vannes, Université Bretagne Sud

minh-tan.pham@univ-ubs.fr



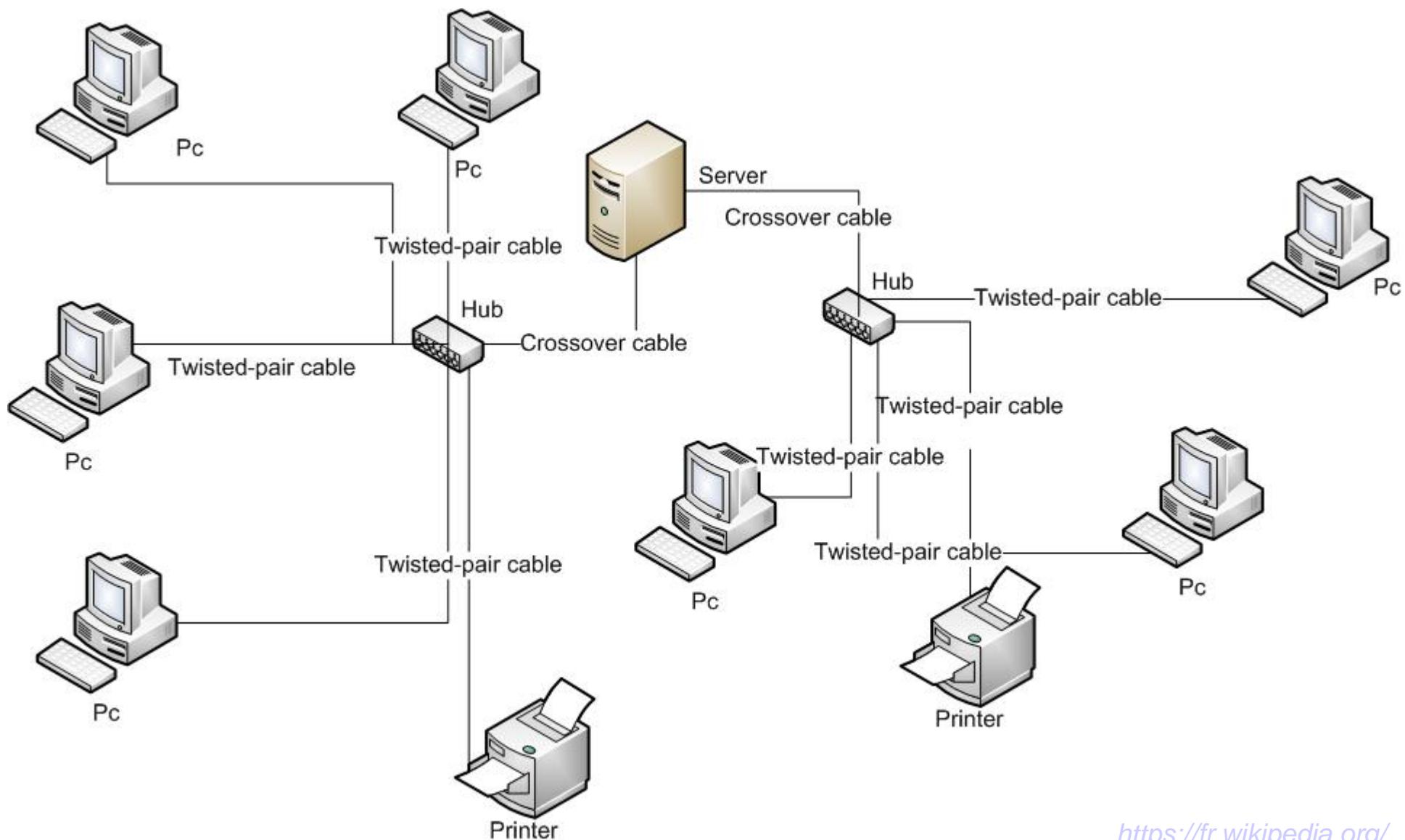
Plan du cours

- 1. Notions de base et un peu d'histoire**
- 2. Représentation de l'information**
- 3. Mesures de performance**
- 4. Domaine d'utilisation des réseaux**
- 5. Structure d'un réseau**
- 6. Architecture des réseaux**
- 7. Les services**
- 8. Organismes de normalisation**

1. Notions de base

- **Informatique:** traitement automatique de l'information
- **Réseau d'ordinateurs:** un ensemble d'ordinateurs interconnectés (échanger des informations) par des supports de transmission
- **Réseau informatique:** assurer les communications entre les ressources informatiques, y compris:
 - Matériels:
 - Composants de traitement: ordinateurs, imprimantes, scanners, etc.
 - Composants de transmission: câbles, modems, cartes réseaux, routeurs, commutateurs (Switch), concentrateur (Hub), etc.
 - Logiciels: applications, jeux, base de données, etc.

1. Exemple d'un réseau informatique



<https://fr.wikipedia.org/>

1. Un peu d'histoire

- **Les progrès des télécommunications**
 - En terme de débit: nombre de bits transportés
 - En terme de la qualité des services offerts
 - En terme de coût
- **Les progrès des ordinateurs**
 - En terme de vitesse des processeurs
 - En terme de capacité des mémoires
 - En terme de coût
- **Réseaux et systèmes distribués**

2. Représentation de l'information

- **Nature de l'information à transmettre:** suite d'éléments binaires codant l'information (base 2: chiffres 0 et 1)
- **Transmission des informations:** se fait en modifiant l'état logique de la voie
- **Unités utilisées:**
 - Bit (binary digit):
 - 1 bit → coder 2 états: 0 et 1
 - 2 bits → coder 4 états: 00, 01, 10 et 11
 - n bits → coder 2^n états
 - Octet : une suite de 8 bits → 1 octet = 8 bits

2. Représentation de l'information

- Unités multiples des bits et octets (attention!):

Unité	Symbol	Valeur (bits)
kilo-bit	Kb	10^3
méga-bit	Mb	10^6
giga-bit	Gb	10^9
téra-bit	Tb	10^{12}

Unité	Symbol	Valeur (octets)
kibi-octet	Kio	$2^{10} = 1024$
mébi-octet	Mio	2^{20}
gibi-octet	Gio	2^{30}
tébi-octet	Tio	2^{40}

➔ TD1 Débit sur les conversions entre octets et bits ! (IMPORTANT)

3. Mesure de performance

- **Débit** : mesure la quantité d'information que le réseau peut transmettre par unité de temps

$$\text{débit} = \frac{\text{quantité d'information}}{\text{temps}}$$

- **Unité**: bit par seconde (*bps* ou *b/s*, ou bs^{-1})
- **Taux d'utilisation**: rapport débit utile/débit nominal
 - *Débit nominal*: quantité théorique maximale d'information pouvant être transmise par unité de temps
 - *Débit utile*: quantité d'information effectivement transmise par unité de temps

3. Mesure de performance

$$\text{débit} = \frac{\text{quantité d'information}}{\text{temps}}$$

Unités de débit binaire			v · d · m
Système international (SI)			
Unité	Notation	Valeur	
bit par seconde	bit/s ou b/s ou bps	1 bit/s	
kilobit par seconde	kbit/s ou kb/s	10^3 bit/s	
mégabit par seconde	Mbit/s ou Mb/s	10^6 bit/s	
gigabit par seconde	Gbit/s ou Gb/s	10^9 bit/s	
térabit par seconde	Tbit/s ou Tb/s	10^{12} bit/s	
pétabit par seconde	Pbit/s	10^{15} bit/s	
exabit par seconde	Ebit/s	10^{18} bit/s	
zettabit par seconde	Zbit/s	10^{21} bit/s	
yottabit par seconde	Ybit/s	10^{24} bit/s	

<https://fr.wikipedia.org/>

3. Mesure de performance

- **Exemples de débit** (source: Wikipédia)

- *ARCnet* : 2,5 Mbit/s, 20 Mbit/s
- *Token ring* : 4 Mbit/s, 16 Mbit/s
- *Ethernet* : 10 Mbit/s
- *Fast Ethernet* : 100 Mbit/s
- *Fibre Channel* : 4 Gbit/s, 8 Gbit/s

- **Temps d'acheminement des message :**

- *Temps de transmission*: le temps mis pour transmettre la quantité d'information du message
- *Temps de propagation*: le temps mis pour que le signal se propage sur le matériel (plus les retards introduits par des matériels)
- *Temps total = temps de transmission + temps de propagation*

4. Domaine d'utilisation des réseaux

- **Finalité des réseaux:**

- Permettre le partage des ressources
- Accroître la résistance aux pannes
- Diminuer les coûts

- **Applications:**

- Accès à des services à distance: base de données, applications web, programme, etc.
- Communication: mail, news, talks, visioconférence, etc.

4. Catégories des réseaux

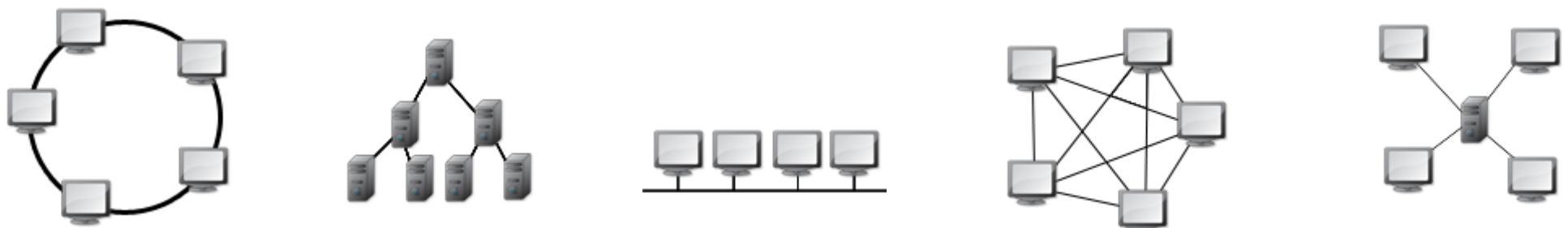
Nom	Distance	Débit
Personal Area Network (PAN)	quelques mètres	<i>1Mbps</i>
Local Area Network (LAN)	jusqu'à 2km	<i>de 10Mbps à 1Gbps</i>
Metropolitan Area Network (MAN)	jusqu'à 100km	<i>~ 100Mbps</i>
Wide Area Network (WAN)	milliers de km	<i>quelques Mbps</i>

■ Exemples:

- PAN chez un particulier
- LAN dans un bâtiment, un site industriel
- MAN à l'échelle d'une agglomération, une ville
- WAN à l'échelle d'un pays

5. Structure d'un réseau

- **Topologie de réseaux:** organisation structurelle qui précise comment sont interconnectés les différents composants
 - Bus
 - Étoile
 - Anneau
 - Arbre
 - Graphe (topologie complètement/partiellement maillée)



5. Structure d'un réseau

▪ Canaux de communication point à point

- *Principe de communication:* pour aller d'un équipement terminal à un autre, un message peut traverser plusieurs nœuds de commutation selon le principe stocker renvoyer
- Les nœuds de commutation sont des calculateurs distincts des équipement terminaux
- Exemple: étoile, anneau, arbre, graphe

▪ Canaux de communication à diffusion

- *Principe de communication:* tous les nœuds de commutation reçoivent le message expédié par un équipement terminal
- Un nœud de commutation est un circuit dans l'équipement terminal
- Exemple: bus, satellite ou radio, anneau

6. Architecture des réseaux

- **Modèle des couches**

- Préciser comment les différentes activités sont organisées entre elles
- Réduire la complexité de conception, de réalisation et d'implémentation des interactions entre les différentes activités

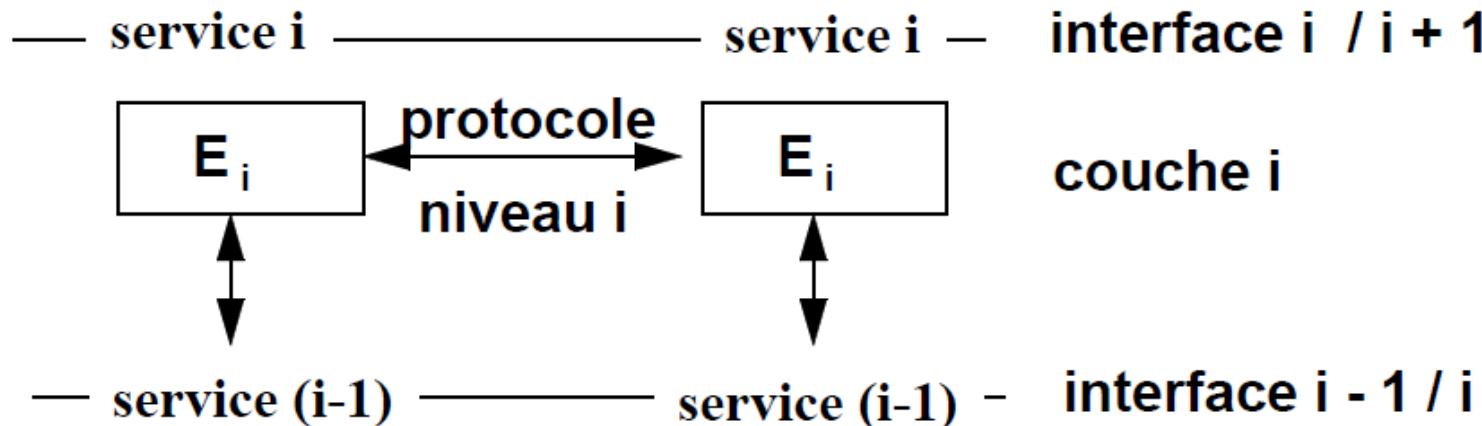
- **Couche** (niveau)

- Ensemble des entités formant un sous-système
- Ne peut dialoguer qu'avec une couche de même niveau sur une autre machine
- S'appuie sur les services offerts par la couche inférieure
- Offre des services à la couche supérieure

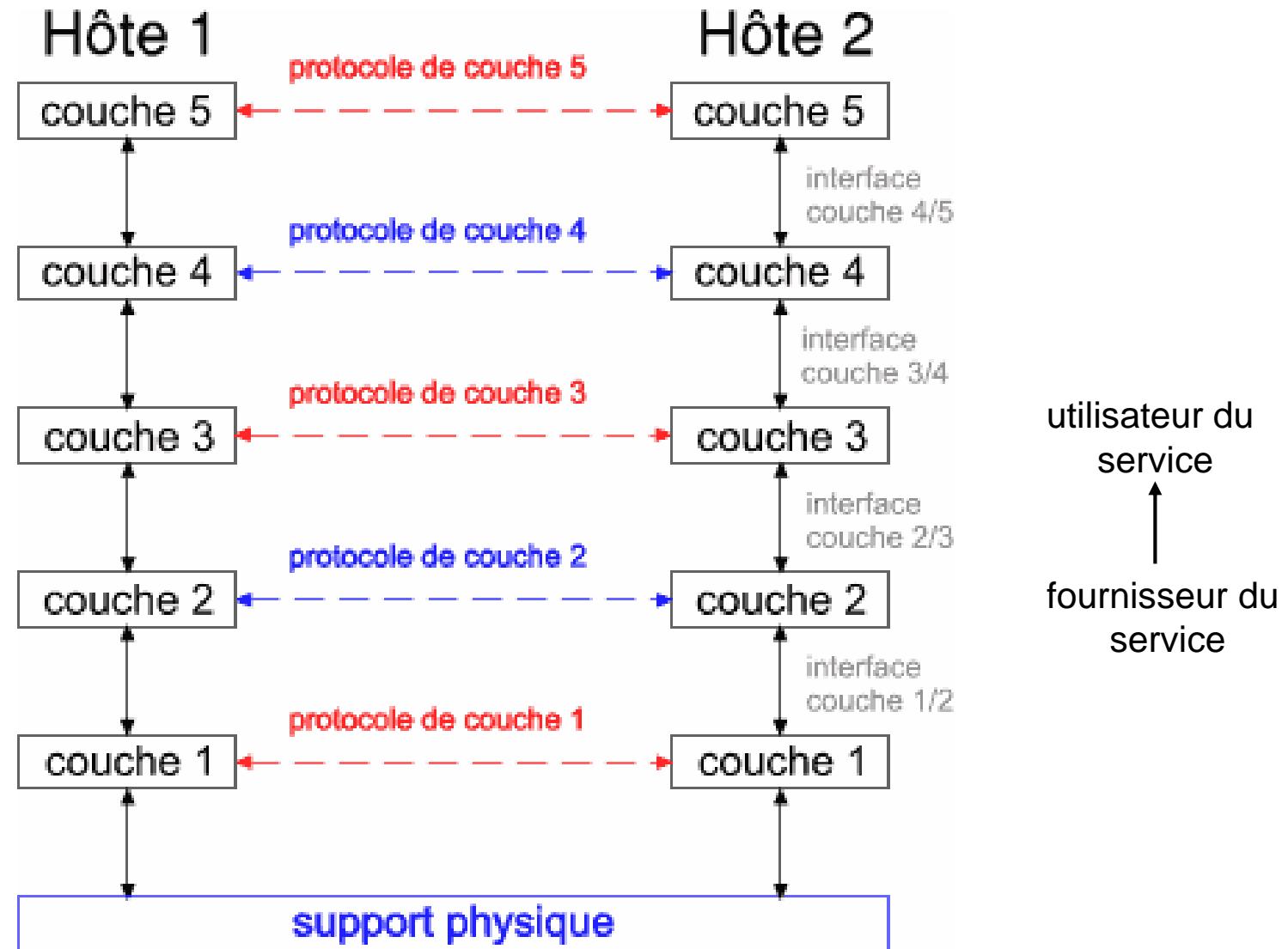
6. Architecture des réseaux

■ Définitions

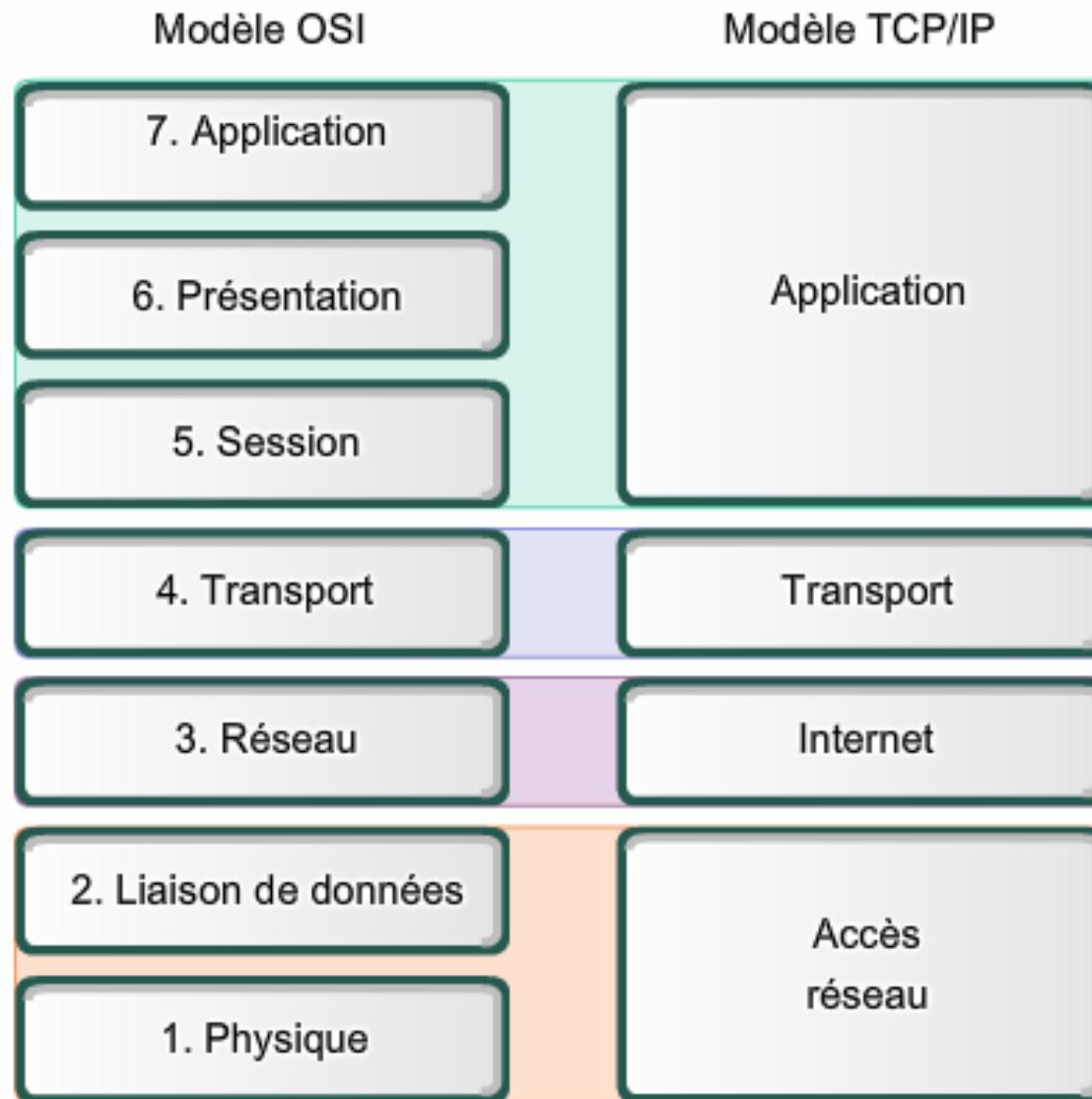
- *Service*: ensemble des fonctions offertes par une couche de niveau i à la couche de niveau $i+1$ d'une même machine
- *Protocole*: ensemble des conventions qui règlent les échanges entre les entités de même niveau qui coopèrent pour rendre un service
- *Interface*: ensemble des règles et des moyens physiques nécessaires pour accéder à un service (*que la couche inférieure offre à la couche supérieure*)



6. Architecture des réseaux



6. Architecture des réseaux



<https://fr.wikibooks.org/>

7. Services: types de services

- **Service orienté connexion: 3 phases de communication**

- Établissement ou ouverture de la connexion
- Transfert des données (communication)
- Fermeture de la connexion

Exemple: le téléphone

- **Service sans connexion**

- On envoie et on reçoit sans préalable

Exemple: le courrier

7. Services: types de services

▪ Qualité de service

Mode	Qualité	Exemple
Connecté	Fiable	Envoi d'un fichier binaire
	Non fiable	Conversation téléphonique
Sans connexion	Fiable : avec acquittement	Achat à distance
	Non fiable: sans acquittement	Envoi d'un message électronique

– R2.04 – Communication et fonctionnement bas niveau –

Cours 2: Modèle OSI - La couche physique

Minh Tan PHAM

BUT INFO, 2022-2023

IUT de Vannes, Université Bretagne Sud

minh-tan.pham@univ-ubs.fr

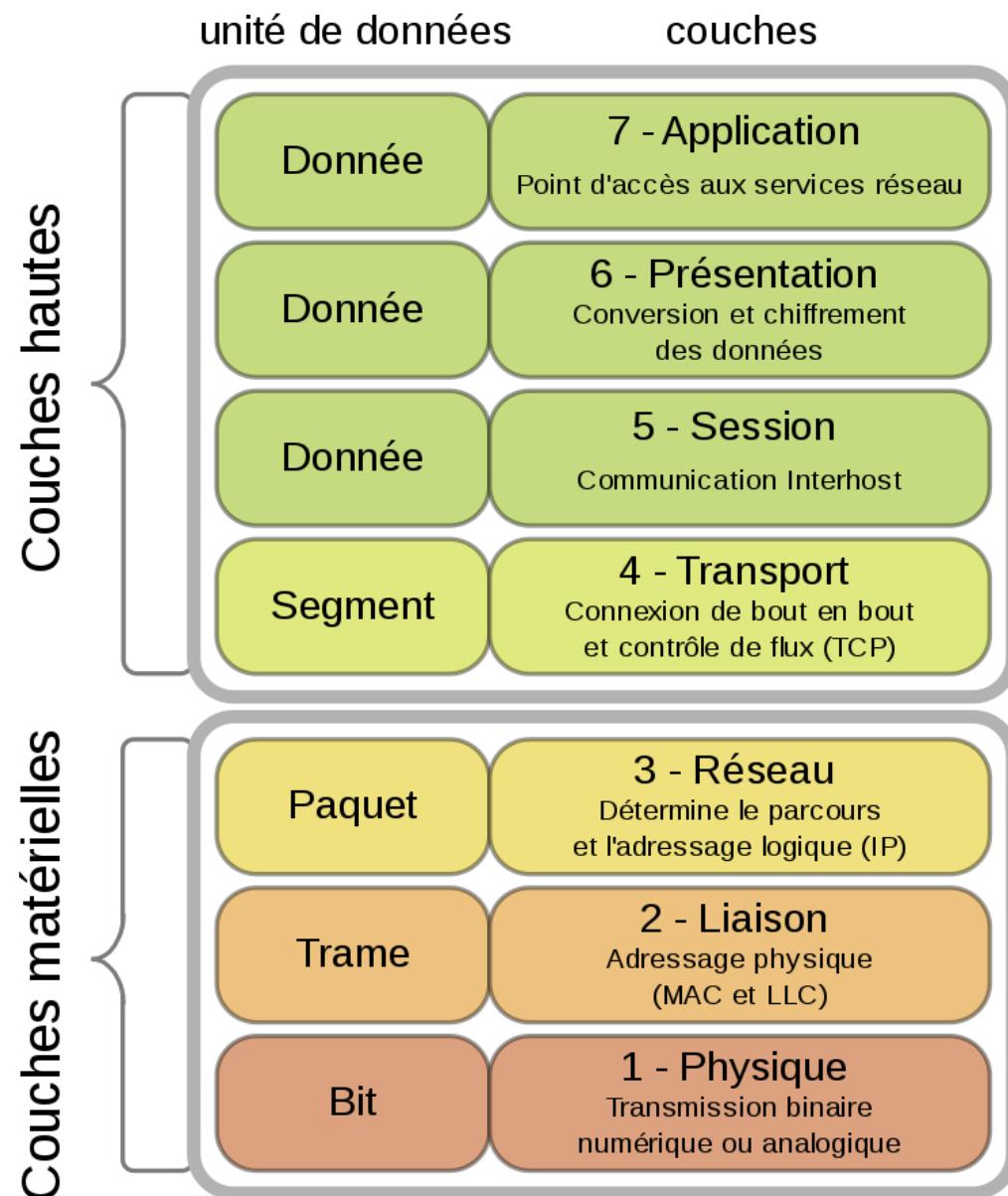


Plan du cours

- 1. Le modèle OSI : rappel**
- 2. Principe de transmission**
- 3. Transmission en mode de base**
- 4. Transmission par modulation**
- 5. Supports de transmission (*lecture comp.*)**

1. Le modèle OSI : rappel

OSI : Open System Interconnection



<https://fr.wikipedia.org/>

1. Le modèle OSI : rappel

- **Physique** : transmet des signaux sous forme numérique ou analogique entre les interlocuteurs par des moyens mécaniques, électriques et fonctionnels
- **Liaison de données** : transforme transmission brute en transmission sans erreur des blocs d'information (trame)
- **Réseau** : Détermine le parcours des données et l'adressage logique (routage et adressage des paquets)
- **Transport** : assure le transport de l'information de bout en bout entre deux machines et le contrôle de flux
- **Session** : contrôle/synchronise le dialogue entre les machines qui communiquent
- **Présentation** : réalise la compression et le chiffrement; vérifie la syntaxe des données changées
- **Application** : offre aux utilisateurs des services normalisés pour la conception de leurs applications

1. Le modèle OSI : rappel

Couche	Résumé
Physique	<i>Comment transmettre des bits sur un support physique?</i>
Liaison de données	<i>Comment transmettre des trames binaires sans erreur?</i>
Réseau	<i>Comment faire transiter les trames sur le réseau?</i>
Transport	<i>Quel type de transmission choisir pour envoyer en message à un destinataire?</i>
Session	<i>Comment mettre en place un dialogue entre deux utilisateurs?</i>
Présentation	<i>Comment faire communiquer des applications de présentations différentes?</i>
Application	<i>Quels sont les outils et services disponibles sur un réseau?</i>

2. Principes de transmission

2. Principes de transmission

- **Nature de l'information à transmettre (rappel)** : suite d'éléments binaires codant l'information (base 2 : chiffres 0 et 1)
- **Transmission des informations** : se fait en modifiant l'état logique E de la voie
- **Correspondance état logique et état physique :**
 - Chaque état logique correspond à une valeur (ou une plage de valeur) de l'état physique)
 - Chaque état logique correspond à une transition entre 2 états physiques
 - Exemple : $E \in [e_1, e_2]$
 - $e_1 = 3.5 \pm 0.5v$ et $e_2 = -3.5 \pm 0.5v$
 - $e_1 = 3.5v \rightarrow -3.5v$ et $e_2 = -3.5v \rightarrow +3.5v$

2. Principe de transmission - Définitions

- **Valence d'une voie** : le nombre d'états logiques distincts utilisés pour représenter l'information. On notera V
 - $V = 2$, voie bivalente $E \in [e_1, e_2]$
 $0 \rightarrow e_1$ et $1 \rightarrow e_2$
 - $V = 4$, voie bivalente $E \in [e_1, e_2, e_3, e_4]$
 $00 \rightarrow e_1 (-2.5v); 01 \rightarrow e_2 (-1v); 10 \rightarrow e_3 (+1v)$ et $11 \rightarrow e_4 (+2.5v)$
- **Moment élémentaire** : durée minimale T_m (en secondes) pendant laquelle une valeur du paramètre physique (appelée symbole) doit être maintenue constante pour être lue par le récepteur

2. Principes de transmission - Définitions

- **Vitesse de modulation** : le nombre de valeurs physiques émises (ou le nombre de changements d'états physiques) par unité de temps. On note R_m (en bauds)

$$R_m = \frac{1}{T_m}$$

- **Débit binaire** : le nombre de valeurs logiques transmises (quantité d'information) par seconde. On le note D (bit/s)

$$D = \frac{R_m}{k} \log_2(V)$$

k : nombre de valeurs physiques nécessaires pour coder un état logique

- **Temps de transmission** : $T_t = L/D$ avec L la longueur (en bits) du message et D le débit binaire (en bit/s)

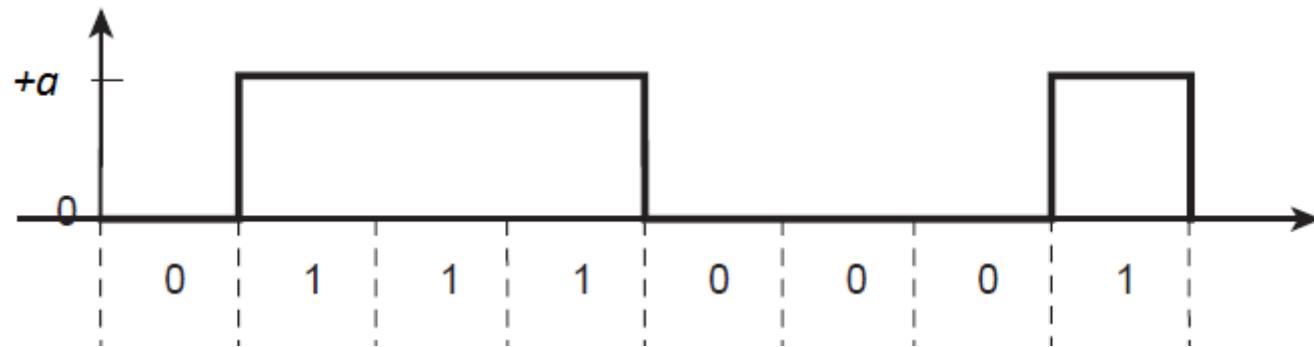
2. Principes de transmission

- **Mode de transmission** : les éléments (suite des bits) sont transmis en série ou en parallèle
- **Mode de base ou par modulation**
 - Transmission en bande de base : les données binaires codées par un signal numérique sont transmises directement sur le câble (*dans les réseaux locaux où les distances entre 2 ordinateurs sont relativement faibles*)
 - Transmission par modulation : transformer les données binaires numériques en un signal analogique (une onde sinusoïdale appelée *porteuse*) qui sera émis sur le câble électrique

3. Transmission en mode de base

3. Transmission en mode de base

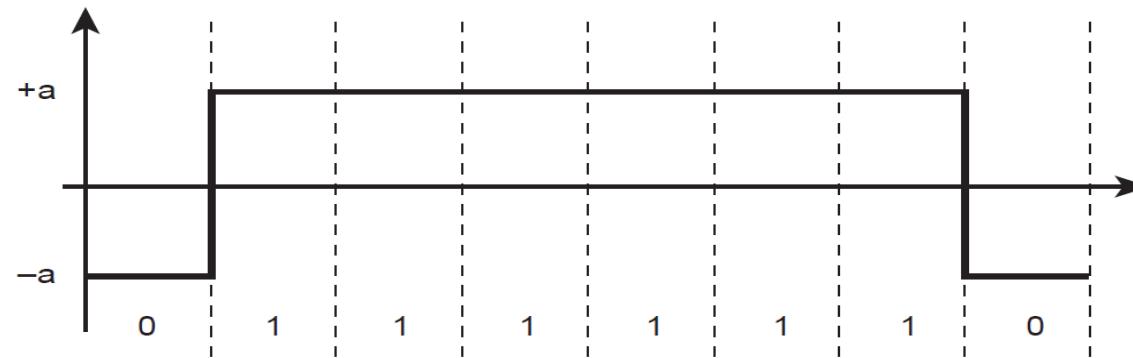
- **Le codage élémentaire** : le bit 1 par un état significatif (tension positive $+a$ volts), le bit 0 par un état de tension nulle



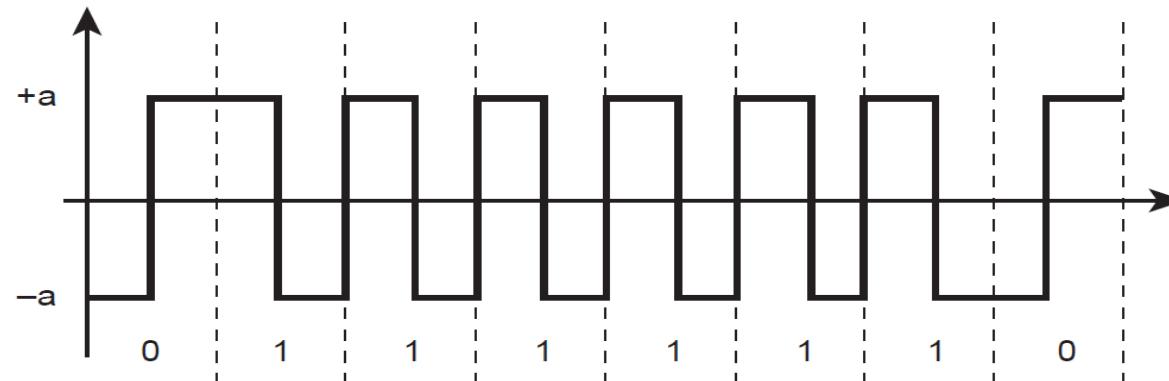
Problème : une tension nulle lue sur le support de transmission correspond à un 0 binaire, mais peut aussi refléter l'absence de donnée transmise !

3. Transmission en mode de base

- **Le code NRZ (No Return to Zero)** : le bit 1 par un état significatif (tension positive $+a$ volts), le bit 0 par état opposé ($-a$ volts)



- **Le code Manchester (code biphasé)** : le bit 1 par un passage de la tension $+a$ vers $-a$ volts et le bit 0 par un passage inverse



3. Transmission en mode de base

■ D'autres types de codage (*TD3 codage*)

■ Code **BIPOLAIRE**

- Bit 1 → tension $+a$ volts ou $-a$ volts, Bit 0 → tension nulle

■ Code **NRZI** (No Return to Zero Inverted)

- Bit 1 → tension $-a$ volts, Bit 0 → tension $+a$ volts

■ Code **Manchester différentiel** (biphase différentiel)

- Bit 0 → transition dans le même sens que la précédente
- Bit 1 → transition dans le sens inverse de la précédente

■ Code de **Miller**

- Bit 1 → transition en milieu de temps horloge
- Bit 0 → absence de transition

3. Transmission en mode de base

■ Exemple de débit binaire :

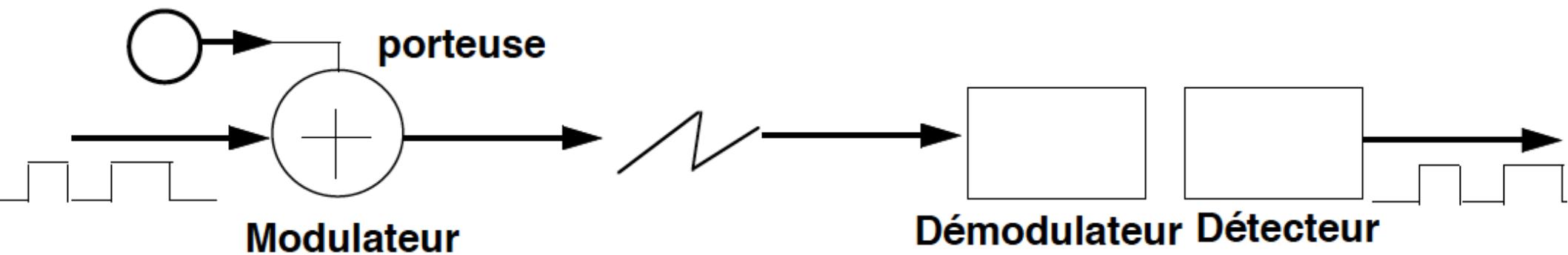
- *Le code NRZ permet de transmettre 1 bit pour une tension donnée (+a volts ou -a volts), d'où k=1*

$$D = \frac{R_m}{k} \log_2(V) = R_m$$

- *Le code de Manchester nécessite de lire 2 valeurs physiques reçues pour connaître l'état logique (0 et 1), d'où k=2*

$$D = \frac{R_m}{k} \log_2(V) = \frac{R_m}{2}$$

4. Transmission par modulation



4. Transmission par modulation

- **Notion de signal** : un signal analogique peut être représenté par une courbe sinusoïdale :

$$y(t) = A \sin(2\pi ft + \varphi)$$

A : amplitude (crête à crête)

f : fréquence (Hz)

φ: phase (décalage de l'onde par rapport à l'origine)

L'information codée sert à modifier un ou plusieurs des paramètres (A, f, φ) de l'onde porteuse

- **Bandé passante** : l'intervalle de fréquences $[f_{min}, f_{max}]$ qu'un support physique accepte et laisse passer. La largeur de la bande passante

$$W = (f_{max} - f_{min})$$

- **Loi de Nyquist** : vitesse de modulation maximale

$$R_{mmax} = 2W$$

4. Transmission par modulation

- **Théorème de Shannon:** débit binaire maximal d'une ligne de transmission qui possède une largeur de bande passante W

$$D_{max} = W \log_2 \left(1 + \frac{S}{B} \right)$$

Le rapport $\frac{S}{B}$: qualité de l'environnement dans lequel est placée la ligne électrique (*quotient de la puissance moyenne du signal S par celle du bruit B*)

On l'appelle aussi capacité théorique de la ligne.

- Remarque : le débit réel est loin de la capacité théorique

4. Transmission par modulation

- Norme V21 : Modulation par saut de fréquence (Frequency shift keying FSK)
 - le bit 0 est représenté par la fréquence f_1
 - le bit 1 est représenté par la fréquence f_2
 - $f_1 = f_0 - x$ et $f_2 = f_0 + x$ avec f_0 la fréquence porteuse du signal
- La norme V21 définit ainsi des modems full-duplex avec deux porteuses sur 2 canaux de transmission :

$$\text{canal 1} \begin{cases} f_1 = f_0 - x \\ f_2 = f_0 + x \end{cases}$$

$$\text{canal 2} \begin{cases} f'_1 = f'_0 - x \\ f'_2 = f'_0 + x \end{cases}$$

La norme V21 du CCITT (IUT-T)

Voie 1	
bit	fréquence
0	1180 Hz
1	980 Hz

Voie 2	
bit	fréquence
0	1850 Hz
1	1650 Hz

4. Transmission par modulation

- Norme V22 : Modulation par saut de phase (Phase shift keying PSK)

- le bit 0 est représenté par la phase φ_1
- le bit 1 est représenté par la phase φ_2
- φ_1 et φ_2 sont choisies de façon très différentes pour éviter les erreurs de transmission

La norme V22 du CCITT (IUT-T) pour des valeurs logiques de 1 ou 2 bits par phase :

1 bit par état de phase	
bit	phase
0	90°
1	270°

2 bits par état de phase	
bit	fréquence
00	0°
01	90°
10	180°
11	270°

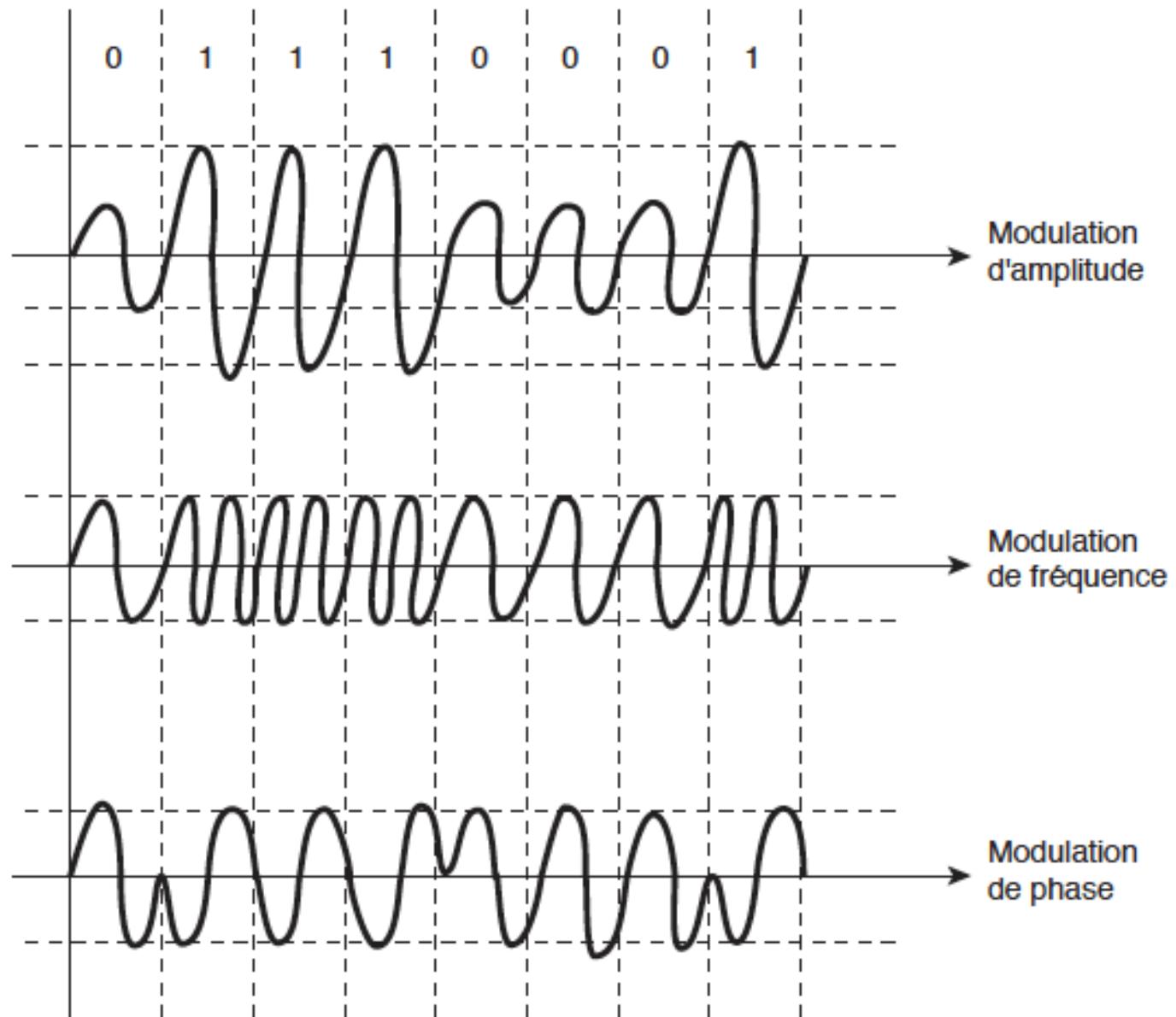
4. Transmission par modulation

- Norme V29 : Modulation par saut de phase et amplitude (Phase shift keying/Amplitude Modulation PSK/AM)
 - PSK avec 8 états de phase
 - Modulation d'amplitude sur 2 valeurs
 - Soit, 16 valeurs logiques codées sur une seule valeur physique

(TD3 codage)

4. Transmission par modulation

Exemples de modulations simples



4. Transmission par modulation

▪ Propriétés des systèmes de modulation

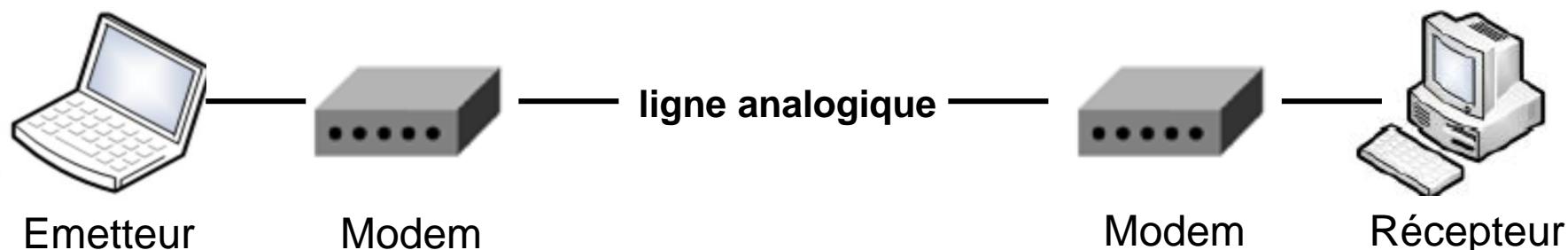
- Saut de phase et d'amplitude exige une largeur de bande (W) plus faible que le saut de fréquence, mais une puissance plus élevée pour un même taux erreur
- FSK : modulations à 4 ou 8 états → réduire le taux d'erreur à puissance constante au prix d'une plus grande largeur de bande
- PSK ou AM : modulations à 4 ou 8 états → réduire la largeur de bande, à taux d'erreur constant, au prix d'une puissance plus élevée

▪ Domaines d'utilisation

- Saut de fréquence : équipement à faible vitesse ou voies à très large bande passante
- Saut de phase et d'amplitude : souvent combinés pour des transmissions à grande vitesse

4. Transmission par modulation

- **Modem** : élément actif qui permet d'utiliser une ligne physique en modulation
- **Rôles :**
 - Modulation et démodulation : conversion de données numériques ↔ signaux analogiques
 - Ajustement du signal analogique : avant émission, le signal analogique est adapté au câble → mieux utiliser la bande passante + minimiser les pertes et erreurs
 - Gestion de la synchronisation du récepteur avec l'émetteur
 - mode synchrone
 - mode asynchrone



4. Transmission par modulation

■ Méthode synchrone/asynchrone :

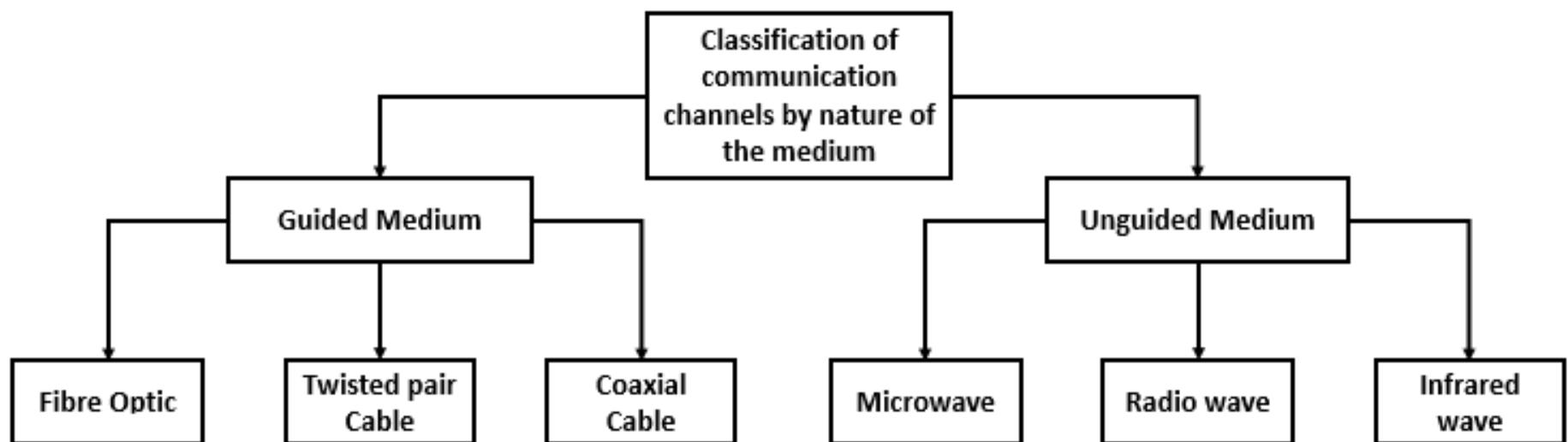
- **synchrone** : émetteur et récepteur disposent d'un même référentiel temporel (horloge) qui détermine les instants significatifs de dépôt et de prélèvement des bits

→ *utilisé pour des transmissions rapides, à grande distance et qui mettent en jeu de grandes quantités d'information*

- **asynchrone** : l'émission est commencée à des instants choisis arbitrairement. Les horloges de l'émetteur et du récepteur ont la même fréquence. L'horloge bit du récepteur est définie à partir de début de bloc

→ *utilisé pour des liaisons courtes à basse vitesse où la source de données produit des caractères à des instants aléatoires*

5. Supports de transmission (lecture complémentaire)

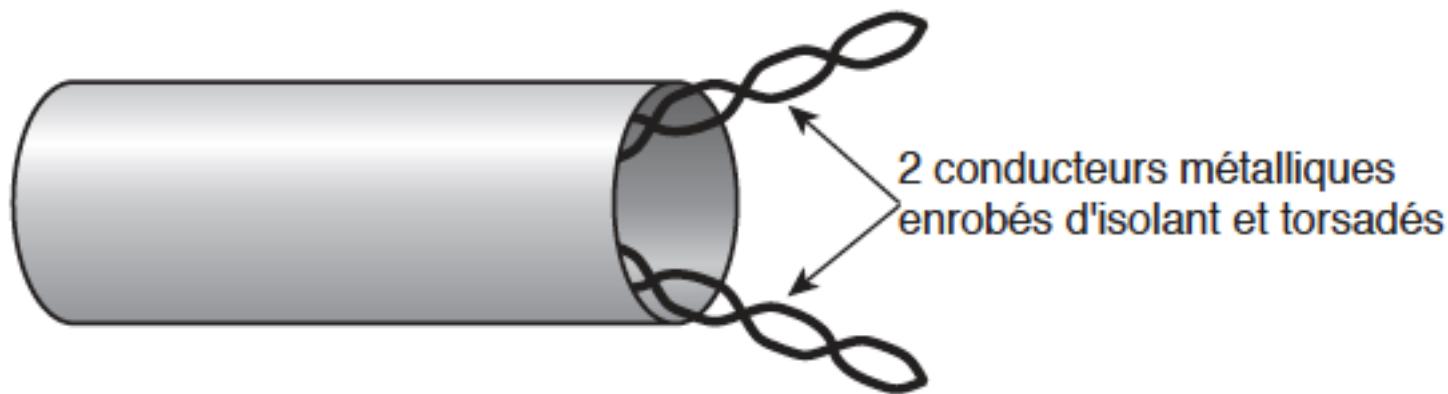


<https://fr.wikipedia.org/>

5. Supports de transmission

■ Câble électrique à paires torsadées (Twisted Pair)

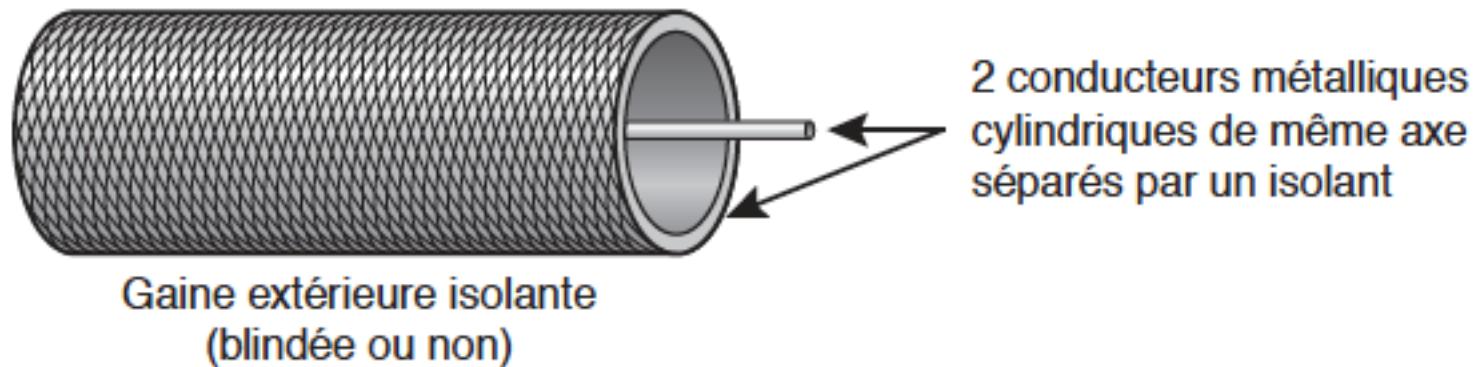
- support physique le plus utilisés sur le marché
- cas d'usage : connexion d'un poste de travail au concentrateur du réseau; interconnexion d'éléments actifs (concentrateurs, commutateurs, relanceurs, etc.)
- il est constitué de plusieurs fils de cuivre torsadés par paires
- Connecteurs appropriés : RJ45 pour 4 paires ou RJ11 pour 2 paires



5. Supports de transmission

■ Câbles coaxiaux (Coaxial cable)

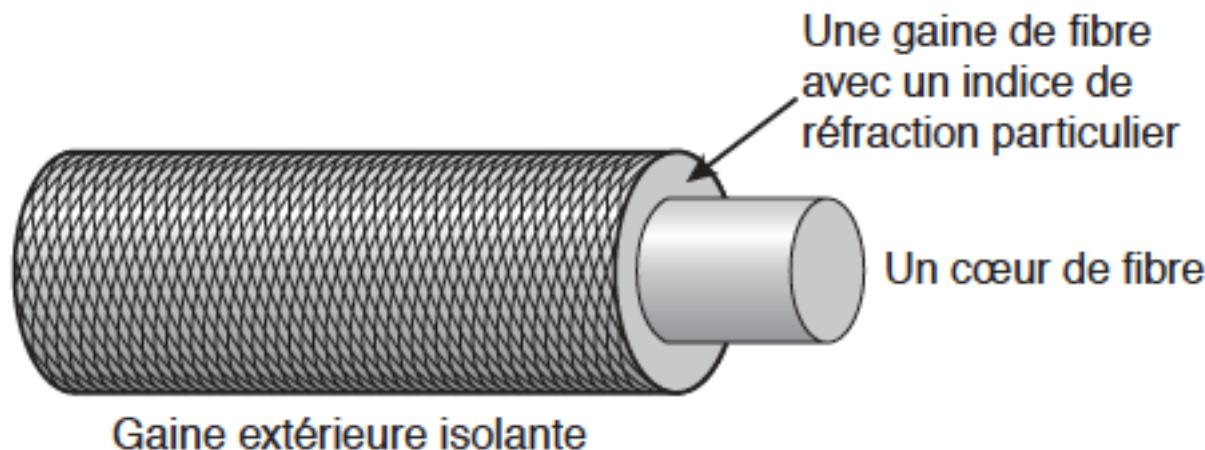
- câble électrique constitué de deux conducteurs métalliques cylindriques de même axe séparées par un isolant
- meilleure performance que la paire torsadée (affaiblissement moindre, transmission de signaux de fréquences plus élevées, etc.)
- sur 1 km, débit → plusieurs dizaines de Mbit/s; distance > 10 km, débit < 10 kbit/s



5. Supports de transmission

■ Fibres optiques (Fiber optic)

- cylindre constitué d'un matériau conduisant la lumière, enveloppé dans un isolant
- émetteur: diode électroluminescente ou laser ; récepteur: photodiode
- avantages: diamètre extérieure ordre 0.1 mm; poids quelques grammes au km
- réseau de haut ou très haut débit: sur des distances de plusieurs kms → débit de 1 à plusieurs centaines de Gbit/s



5. Supports de transmission

▪ Réseaux sans fil (ondes électromagnétiques)

- Espace hertzien (autorisent la transmission des ondes électromagnétiques): l'air, le vide...
- Onde radio : 10 kHz → 300 GHz. Débit 2 à plus de 20 Mbit/s (distance < 20 km)
 - Voies radios 10kHz → 1GHz : utilisées en diffusion (très peu pour transfert de données sauf applications militaires)
 - Faisceaux hertzien : utilisent les fréquences très élevées (2GHz jusqu'à 40 GHz) pour la transmission par satellite géostationnaire (orbite 36000km), pour celle des chaînes télévisions ou pour celle dans les réseaux téléphoniques de longue distance. Débit quelques centaines de Mbit/s
 - Ondes infrarouges, ondes visibles : diffusion dans l'espace hertzien est relativement restreinte (les fréquences élevées ne peuvent pas traverser la matière physique)

5. Supports de transmission

▪ Affectation des fréquences en France

Gamme de fréquences	Type d'utilisation
10 kHz – 150 kHz	Communications radiotélégraphiques
150 kHz – 300 kHz	Radiodiffusion (grandes ondes)
510 kHz – 1605 kHz	Radiodiffusion (petites ondes)
6 MHz – 20 MHz	Radiodiffusion (ondes courtes)
29,7 MHz – 41 MHz	Radiotéléphonie
47 MHz – 68 MHz	Télévision
68 MHz – 87,5 MHz	Liaisons radio en modulation de fréquence
87,5 MHz – 108 MHz	Radiodiffusion
108 MHz – 162 MHz	Radiotéléphonie
162 MHz – 216 MHz	Télévision
216 MHz – 470 MHz	Radiotéléphonie
470 MHz – 860 MHz	Télévision et radar
860 MHz – 960 MHz	Radiotéléphonie
Autour de 1 800 MHz	Radiotéléphonie
Entre 6 et 30 GHz	Services satellites en fixe

– R2.04 – Communication et fonctionnement bas niveau –

Cours 3: La couche liaison de données

Minh Tan PHAM

BUT INFO, 2022-2023

IUT de Vannes, Université Bretagne Sud

minh-tan.pham@univ-ubs.fr



Plan du cours

1. Introduction

2. Constitution des trames

3. Commutation

4. Contrôle d'erreur

- Codes détecteurs
- Codes correcteurs

5. Contrôle de flux et gestion des acquittements

- Protocole attente/réponse
- Transmission avec anticipation

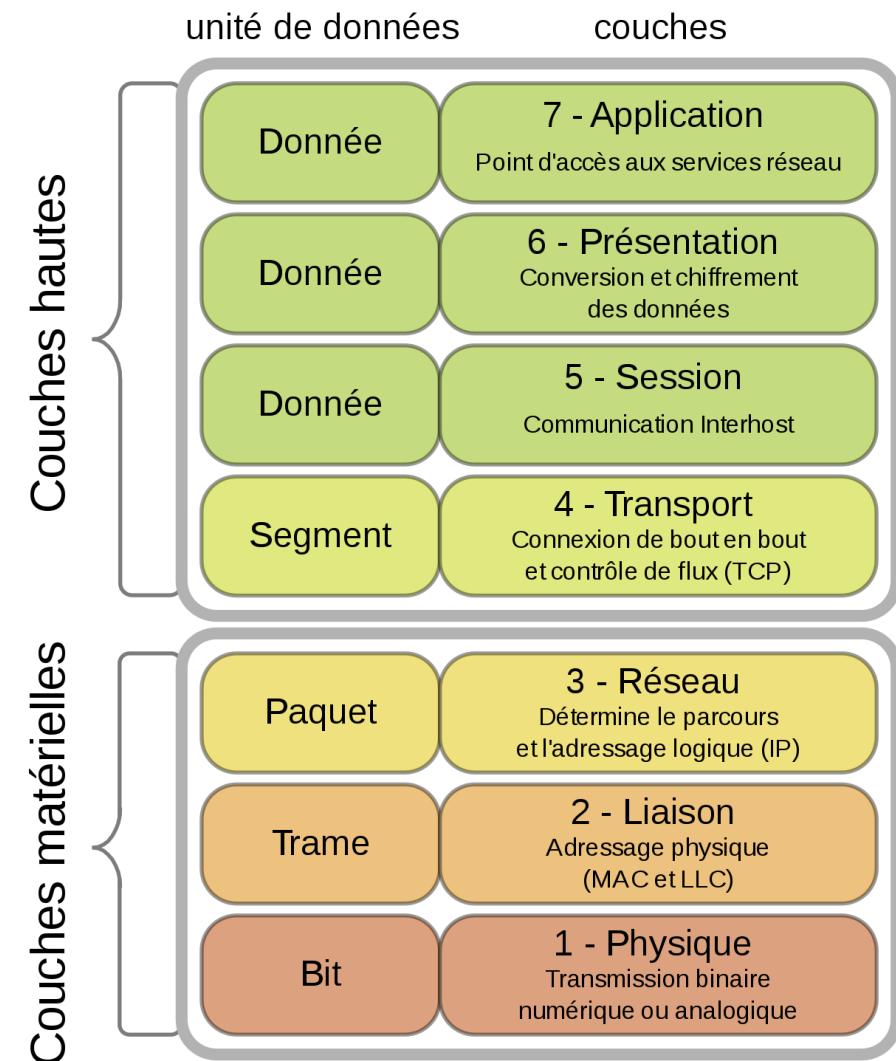
6. Exemple de protocole de liaison de données HDLC/PPP

1. Introduction

Couche liaison de données :

- Ensemble des équipements et des logiciels qui fournissent les moyens fonctionnels nécessaires pour acheminer des données avec un taux d'erreurs garanti
- Dans certains cas, fournit les moyens de détecter et potentiellement corriger les erreurs qui peuvent survenir au niveau de la couche physique

Rappel : Comment transmettre des trames binaires sans erreur ?



<https://fr.wikipedia.org/>

1. Introduction

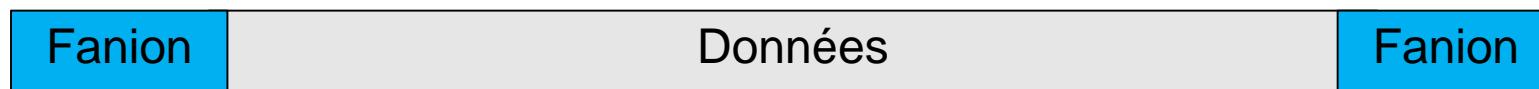
- **Les services fournis à la couche réseau:**

- **Service sans connexion et sans acquittement :** convient si le niveau physique a un taux d'erreur très faible et si on suppose que les erreurs sont corrigées par les niveaux supérieurs. Il est utilisé pour transmettre la parole, des flots temps réel. C'est également le type de service utilisé sur les réseaux locaux
- **Service sans connexion et avec acquittement :** L'utilisateur sait si son message est ou non arrivé. Si la trame n'a pas pu être transmise, il peut l'émettre à nouveau
- **Service avec connexion :** demande un établissement de la connexion, chaque trame envoyée est numérotée. Chaque trame envoyée est reçue une fois et une seule. Les trames sont reçues dans l'ordre où elles ont été envoyées.

2. Constitution des trames

2. Constitution des trames

- **Notion de trame :** Un train de bits émis par la couche physique peut être déformé de trois façons : bits perdus, bits erronés, bits créés accidentellement → fixer une unité de contrôle d'erreur.
- **Techniques pour constituer des trames :**
 - **La taille en bits d'une trame est fixée**
 - **Compter et transmettre la taille de trame au destinataire**
 - **Utiliser des caractères de début et de fin de trame**
 - **Utiliser des fanions pour marquer le début et la fin de la trame :** une suite binaire particulière, connue de l'émetteur et du récepteur, est placée aux deux extrémités de la trame. Le fanion le plus classique : **01111110**.



2. Constitution des trames

▪ Notion de transparence binaire:

- masquer toutes les suites binaires semblables au fanion en insérant lors de l'émission un 0 dès que l'on lit cinq 1 de suite.
- à la réception, on réalise l'opération inverse pour retrouver les données brutes

```
0110111111011111001  
↓  
011011111011011111001  
↓  
01111110 011011111011011111000101111110  
↓  
011011111011011111001  
↓  
0110111111011111001
```

3. Commutation

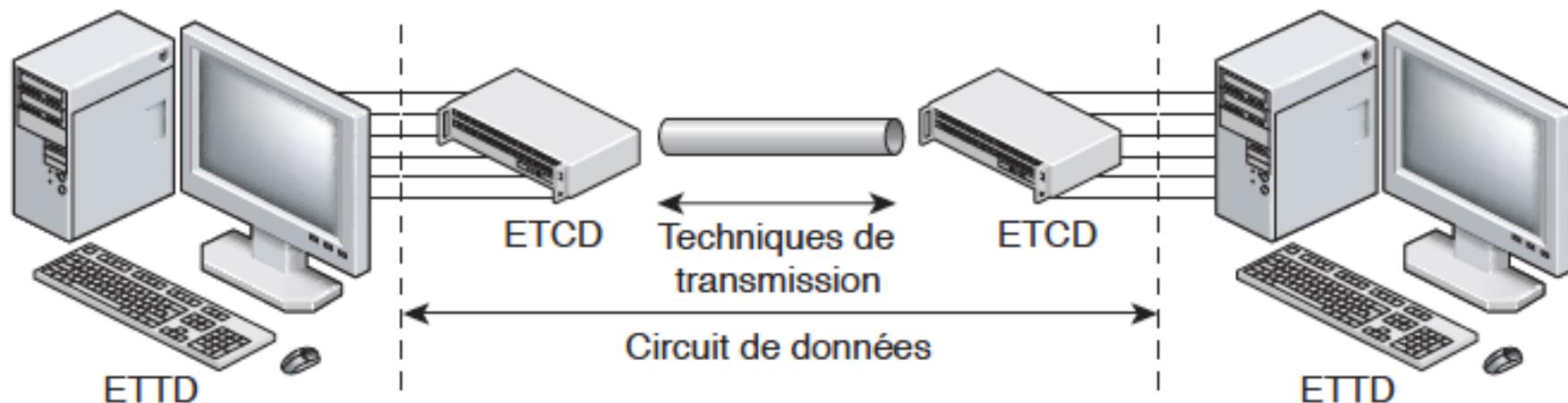
3. Commutation

- **Réseau à commutation** : réseau longue distance qui propose des techniques permettant d'acheminer de manière optimisée des trames de niveau liaison à travers un réseau maillé de communication
- **La commutation** → une famille des techniques de partage d'une voie composée
- **Méthodes de commutation** :
 - Commutation de circuits
 - Commutation de messages
 - Commutation de paquets
 - Commutation temporelle asynchrone

3. Commutation - Définitions

- **ETTD et ETCD:**

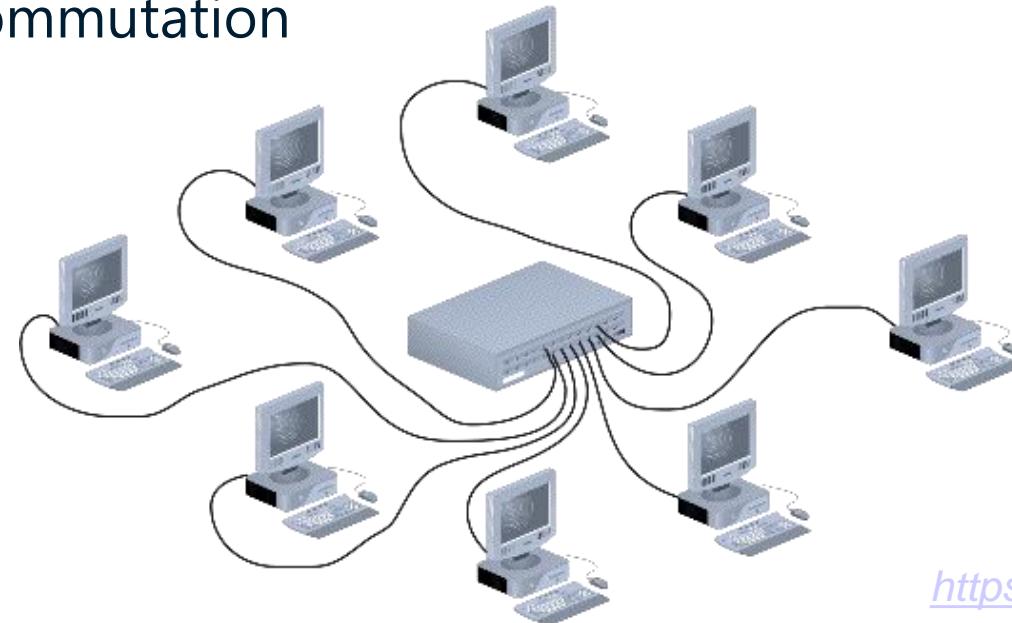
- **ETTD (Equipement Terminal de Traitement de Données)** : élément actif qui agit sur les données elles-mêmes : unité de calcul, ordinateurs serveurs/clients, imprimantes, etc.
- **ETCD (Equipement de Terminaison de Circuit de Données)** : élément permettant la connexion des ETTD au canal de transmission de données : carte de réseau, modem, etc.



3. Commutation - Définitions

■ Commutateur :

- un nœud ayant plusieurs ports de connexion.
- orienter des trames binaires de niveau 2 qu'il reçoit par l'un de ses ports sur un autre port en fonction de la destination recherchée
- $T_c \rightarrow$ le temps de commutation : nécessaire au choix et à la mise en place de la commutation

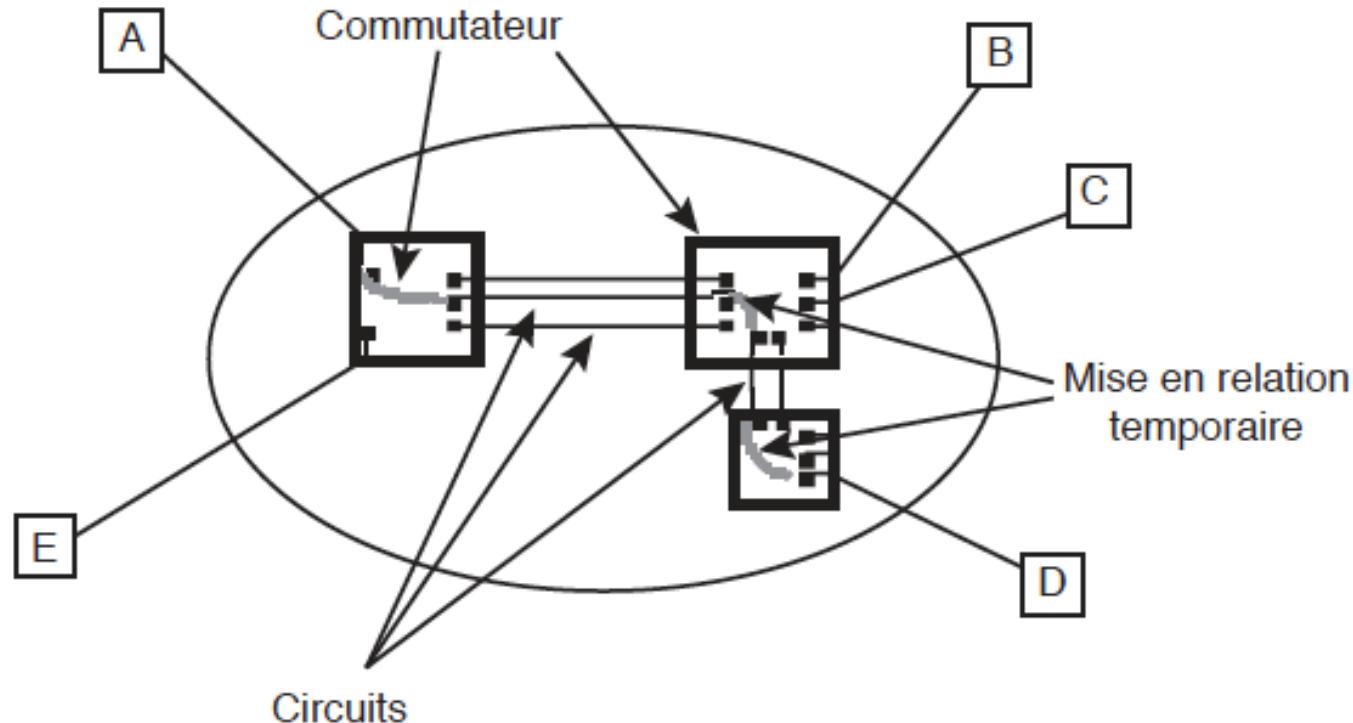


<https://www.supinfo.com/>

3. Méthodes de commutation

■ Commutation de circuits :

- mettre en relation successivement les différents nœuds intermédiaires afin de propager la donnée de l'émetteur au récepteur. La ligne de communication est dédiée à la communication.

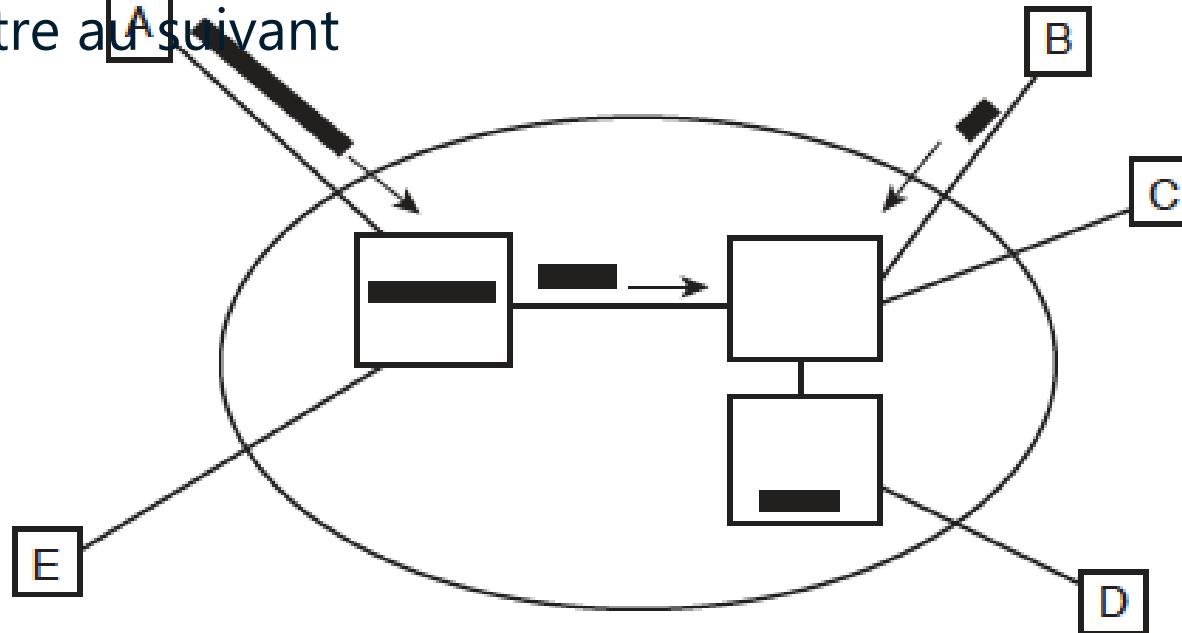


Réservation des ressources physiques pour la durée de la communication
Exemple : réseau téléphonique commuté (RTC)

3. Méthodes de commutation

■ Commutation de messages :

- consiste à transmettre le message séquentiellement d'un nœud à un autre.
- chaque nœud attend d'avoir reçu l'intégralité du message avant de le transmettre au suivant

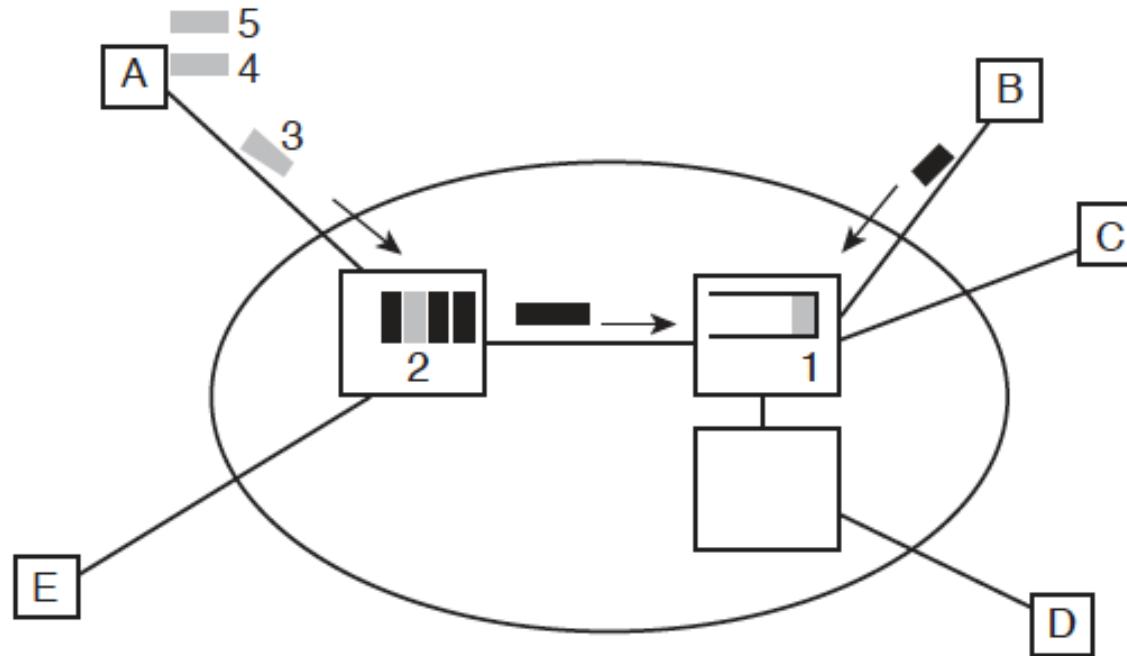


Les liaisons ne sont utilisées que pour la durée de transmission entre équipements adjacents

3. Méthodes de commutation

■ Commutation de paquets :

- consiste à segmenter l'information en paquets de données, transmis indépendamment par les nœuds intermédiaires et réassemblés au récepteur



Le message émis par A est découpé en 5 paquets, acheminés un par un par le réseau.

Il peut y avoir simultanément transmission de plusieurs paquets d'un même message sur différentes liaisons du réseau

3. Méthodes de commutation

▪ **Commutation temporelle asynchrone:**

- Il s'agit toujours de commutation de paquets mais on fait l'hypothèse que leur taille est de quelques dizaines d'octets (faible et fixe).
- Un nœud de commutation n'est plus tenu d'attendre d'avoir reçu tout le paquet pour commencer à le réémettre vers le nœud suivant.
- permet d'atteindre des débits de l'ordre du Gbit

4. Contrôle d'erreur

4. Contrôle d'erreur

Deux techniques pour le contrôle d'erreur :

- **Codes détecteurs** : mettre dans les données juste ce qui est nécessaire pour que le récepteur puisse avec une forte probabilité détecter les erreurs. La correction se fait alors par réexpédition des données erronées
- **Codes correcteurs d'erreurs** : introduire suffisamment de redondance dans les données pour que le récepteur puisse avec une bonne probabilité à la fois détecter et corriger les erreurs

4. Codes de détection

- **Bits de parité :**
 - ajout d'un bit de contrôle à la fin de la chaîne binaire
 - si le nombre de bits 1 est pair → ajouter 0
 - si le nombre de bits 1 est impair → ajout 1
- Remarques :
 - limiter la quantité de données insérées
 - pas fiable : le récepteur ne détecte pas d'erreur si le nombre de bits modifiés (erronés) est pair
 - Si erreur survient sur le bit de parité → la vérification est faussée
 - il existe aussi ***Bits de imparité***

4. Codes de détection

- **Contrôle de parité :**

- parité transversale (Vertical Redundancy Check - VRC)
- parité longitudinale (Longitudinal Redundancy Check - LRC)
- parité croisée (VRC + LRC)

- **Exemple de contrôle de parité croisée par bloc**

caractère1: 10101110 1

caractère2: 10001110 0

caractère3: 10100110 0 ← **VRC**

caractère4: 10100010 1

LRC → 00100100 0 ← bit de parité croisé

VRC: Vertical Redundancy Check

LRC: Longitudinal Redundancy Check.

TD Codage DéTECTeur !!!

4. Codes de détection

- **Code CRC (Code de Redondance Cyclique) :**
 - aussi appelé *Code polynomial* → utilisé dans la majeure partie des LAN
 - on considère que toute chaîne binaire peut être représentée sous une forme polynomiale. Exemple : $10111 \rightarrow 1x^4 + 0x^3 + 1x^2 + 1x + 1$
 - choisir un **polynôme générateur $G(x)$** qui permet le contrôle
- Principe: soit $M(x)$: polynôme associé à la chaîne; $G(x)$: polynôme générateur
 - 1. multiplier $M(x) \times x^r$ où r est le degré de $G(x)$ ↔ ajouter r bits 0 à la fin
 - 2. diviser $M(x) \times x^r$ par $G(x)$ et obtenir $R(x)$ le reste de la division
 - 3. calculer et transmettre le mot $M(x) \times x^r - R(x)$ ↔ concaténer le reste de la division à la chaîne (appelé *champ de contrôle d'erreur*)
 - 4. à la réception, diviser ce polynôme par $G(x)$. **Si le reste est nul → pas d'erreur**

4. Codes de détection

- **Exemple de gestion d'erreurs par code CRC:**

- chaîne binaire à émettre : 10111, soit $M(x) = x^4 + x^2 + x + 1$
- $G(x)$ choisi : $G(x) = x^3 + 1$, soit 1001
- Degré de $G(x)$ est $r=3$. On multiplie $M(x) \times x^3$, soit polynôme 10111000
- On divise celui par $G(x)$

$$\begin{array}{r} 10111000 \\ \hline 1001 \end{array}$$

$$\begin{array}{r} 101 \\ \hline 10101 \end{array}$$

- le reste $R(x) = x^2 + 1$ qui correspond à 101
- trame émise : 10111101 dont 101 est le champ de contrôle d'erreur
- à réception, s'il n'y a pas d'erreur, la division 10111101 par 1001 présente un reste nul; au contraire, le récepteur demande à l'émetteur de lui retransmettre la trame!

TD Codage DéTECTeur !!!

4. Codes correcteurs

■ Vs codes détecteurs :

- plus complexe
- intéressant lorsqu'une ligne physique est très peu fiable → éviter de nombreuses retransmissions de trames
- pas intéressant si peu d'erreurs surviennent → perte de temps importante

→ ***Code de Hamming***

4. Code correcteur de Hamming

- **Mot de code** : si une trame contient m bits de données et r bits de contrôle, on appelle mot du code le mot formé par les $m+r$ bits. On pose $n = m+r$.

message	bits de contrôle	mot du code
0000	000	0000 000
0001	101	0001 101
0010	111	0010 111
0011	010	0011 010

- **Distance de Hamming** : le nombre de bits différents entre deux mots de code

$$d(1100110, 1010110) = 2.$$

- **Distance de Hamming du code complet** : soit M l'ensemble des 2^m mots de codes possibles si on admet que les r bits de contrôle sont calculés en fonction des m bits de données.

$$d(M) = \{\min d(x_1, x_2); x_1, x_2 \in M\}$$

4. Code correcteur de Hamming

- **Propriétés:** pour détecter d erreurs il suffit que la distance de Hamming soit $d + 1$; pour corriger d erreurs il suffit que la distance de Hamming soit $2d + 1$
- Un code de distance $d(M)$ détecte $d(M) - 1$ erreurs et corrige $k = \frac{[d(M)-1]}{2}$ erreurs
- **Exemple de code détecteur :** bit de parité $m=2$, $r=1$.
 $M = \{000, 011, 101, 110\} \rightarrow d(M) = 2$, on détecte 1 erreur
- **Exemple de code correcteur :** $m = 2$, $r = 3$
 $M = \{00\ 000, 01\ 011, 10\ 110, 11\ 101\} \rightarrow d(M) = 3$, on corrige 1 erreur

– R2.04 – Communication et fonctionnement bas niveau –

Cours 4: La sous-couche MAC

Minh Tan PHAM

BUT INFO, 2022-2023

IUT de Vannes, Université Bretagne Sud

minh-tan.pham@univ-ubs.fr



Plan du cours

1. Introduction

2. Méthodes d'accès au support

- Méthodes sans collision
- Méthodes avec collisions (ALOHA, CSMA, CSMA/CD)

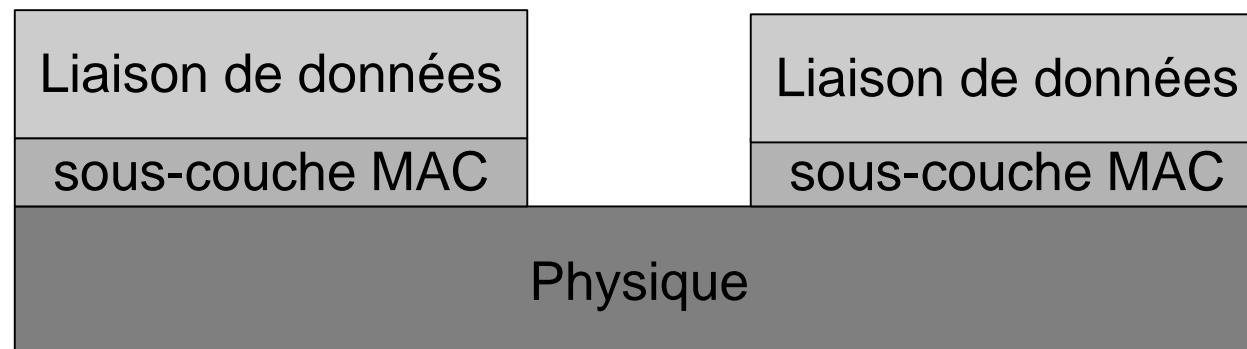
3. Les normes de réseaux

- Ethernet IEEE 802.3
- Token Ring (anneau à jeton) IEEE 802.5
- Wifi 802.11

4. Les ponts (pare-feu)

1. Introduction

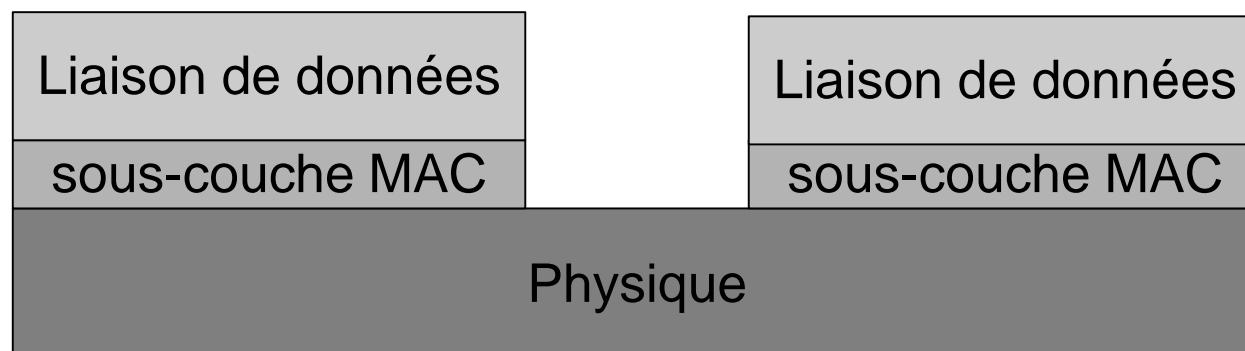
Sous-couche MAC (Medium Access Control) : regrouper toutes les fonctions de niveau liaison de données chargées du contrôle d'accès au support (physique) → éviter les collisions et les erreurs transmission



1. Introduction

Sous-couche MAC (Medium Access Control)

- Différentes méthodes d'accès au support
- Normes de réseaux définis par l'**IEEE** (Institute of Electrical and Electronics Engineers) à partir des méthodes d'accès présentées.



2. Méthodes d'accès au support

2. Méthodes d'accès au support

- Deux familles :
 - Méthodes **sans collision** → traitent le problème en amont, c'est à dire sans collision de données sur le support
 - Méthodes **avec collisions** → solutionnent le problème en aval en réagissant aux collisions de données observées

2. Méthodes d'accès sans collision

- **Allocation centralisée :**

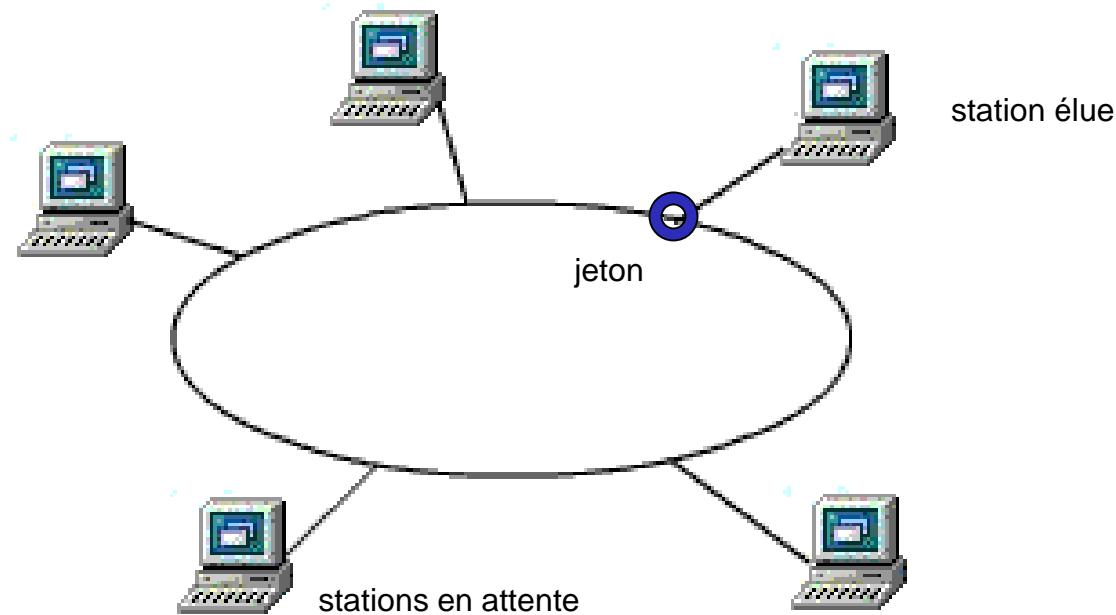
- Une station, le contrôleur, est distinguée et s'occupe d'exécuter
- Le contrôleur interroge régulièrement les stations à tour de rôle pour savoir si elles ont quelque chose à émettre.
- Une station qui a une trame à émettre → soumet sa requête au contrôleur → transmet son message sur la ligne d'entrée et termine avec une indication au contrôleur lui indiquant qu'elle a terminé



2. Méthodes d'accès sans collision

- **Allocation distribuée :**

- autorisation d'émettre des données par un jeton qui circule dans le réseau
- méthode simple et aisée à mettre en œuvre
- méthode la plus répandue dans les premiers LANs (pas aujourd'hui)

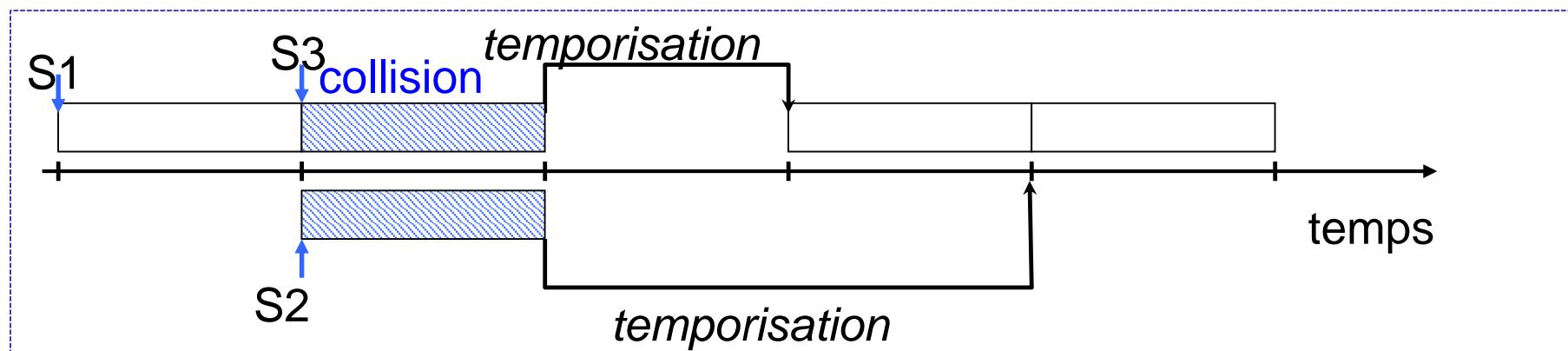
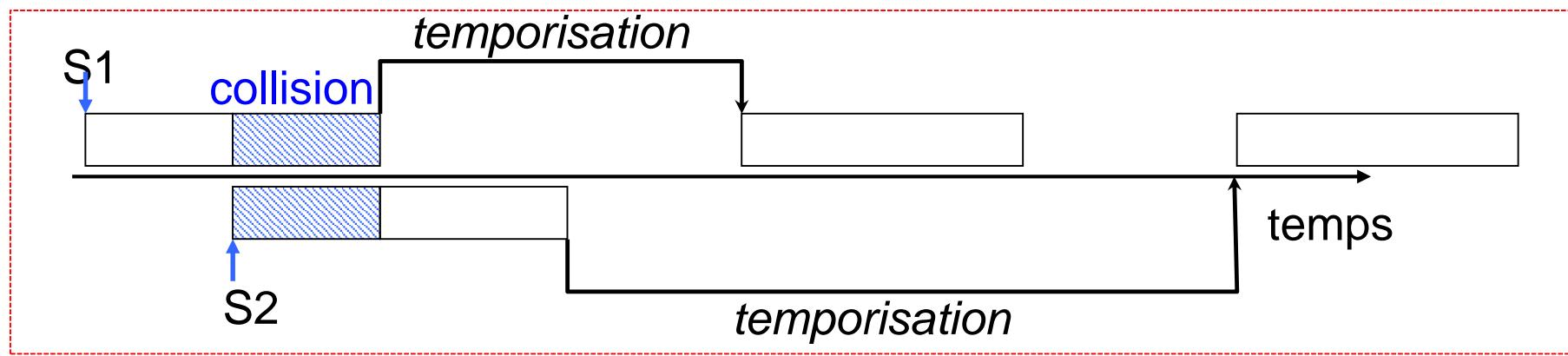


2. Méthodes d'accès avec collisions

- **Méthode ALOHA :** une station veut transmettre des données
 - émettre sur le support sans aucune action préalable
 - écouter le support pour savoir s'il y a eu une collision, si oui attendre un temps aléatoire pour émettre à nouveau
- **ALOHA discréétisé (slotted ALOHA):**
 - ne pas autoriser des émissions de données entièrement libres
 - découper le temps en intervalles réguliers avec horloges synchronisées
 - Un élément veut émettre → attend d'un signal horloge

2. Méthodes d'accès avec collisions

- ALOHA vs ALOHA discrétré



→ Réduit les possibilités de collisions puisqu'on n'a plus de trames qui sont partiellement en collision

2. Méthodes d'accès avec collisions

- **CSMA (Carrier Sense Multiple Access)** : principe comme ALOHA avec **écoute du support (avant l'émission)** → réduire du nombre de collisions
- **Plusieurs variantes:**
 - **CSMA persistant** : Canal occupé, station maintient son écoute jusqu'à une libération du canal. Canal libre, station émet sa trame.
 - **CSMA non persistant** : Si canal occupé, la station ne reste pas en écoute, mais attend une durée aléatoire avant une nouvelle tentative d'envoi
 - **CSMA p-persistant** : Si canal disponible, transmission avec une probabilité **p**.

2. Méthodes d'accès avec collisions

■ CSMA/CD (Collision Detection)

- une des plus évoluées de la famille CSMA
- basé sur CSMA non persistant avec détection de collision
- si collision détectée → station arrête sa transmission
- réduit le besoin de mécanismes de retransmission complexes : station se rend compte que la trame qu'elle envoie n'arrive pas à destination, elle peut retransmettre automatiquement les trames qui ont subi une collision

2. Méthodes d'accès avec collisions

■ Méthode CSMA/CA (Collision Avoidance)

- Autre nom : RTS/CTS (Request to send/Clear to send)
- station réserve le support par un court échange avec le récepteur. Si cet échange aboutit, les autres n'émettent pas
- conçue pour les réseaux locaux sans fil : l'émission par diffusion hertzienne de la trame de requête facilite la concertation de tous les postes

2. Méthodes d'accès au support

Résumé :

- **Méthodes sans collision**
 - Allocation centralisée → contrôleur
 - Allocation distribuée → jeton
- **Méthodes avec collisions**
 - ALOHA (pur, discrétilisé)
 - CSMA (persistant, p-persistant, non persistant)
 - CSMA/CD
 - CSMA/CA

3. Les normes de réseaux

3. Les normes de réseaux

Méthode d'accès	Norme	Architecture/Technologie
CSMA/CD	802.3	Ethernet
	802.3u	Fast Ethernet (haut débit)
	802.3z	Gigabit Ethernet (très haut débit)
	802.3ae	10 Gigabits Ethernet (très haut débit sur longue distance)
jeton	802.4	Token Bus (bus à jeton)
	802.5	Token Ring (anneau à jeton)
	FDDI	utilisée pour l'interconnexion de réseaux
CSMA/CA	802.11	LAN sans fil (Wifi)
	802.15.1	Bluetooth
	802.15.4	LPWAN (Low power WAN)

3. Ethernet 802.3

Format de trame 802.3

7	1	6	6	2	0 - 1500	0 - 46	4 octets
Préambule	Délimiteur de trame	@ dest	@ source	Longueur données	Données	PAD	Contrôle d'erreur

Préambule : succession de 7 octets 10101010 → permet de synchroniser les horloges des stations

Délimiteur de trame : indique le début des données utiles

Adresse destination et Adresse Source (MAC) : déterminer le récepteur/récepteur

Longueur de données : permet au récepteur d'interpréter les champs Données, PAD et Contrôle d'erreur

Données : longueur variable d'une trame à l'autre

PAD : trame 802.3 entre 64 et 1518 octets → si la quantité de données est trop faible, le PAD est ajouté

Contrôle d'erreur : code de redondance cyclique (CRC) basé sur un polynôme générateur de degré 32

3. Ethernet 802.3

Spécification d'une architecture Ethernet

- **Topologie en bus logique**
- **Supports physiques :** (cf. Cours 2 - Supports de transmission)
 - câble à paires torsadées*
 - câble coaxial*
 - fibre optique*
- **Transmission en bande de base (code Manchester)**
- **Débit de 1 à 10 Mbit/s**
- **Méthode d'accès au support : CSMA/CD**
- **Trame** au format 802.3 de longueur de 64 à 1518 octets

3. Ethernet 802.3

Exemples des classes Ethernet 10Mbit/s

	10Base5	10Base2	10BaseT	10BaseF
Débit	10 Mbit/s	10 Mbit/s	10 Mbit/s	10 Mbit/s
Support (câble)	coaxial épais	coaxial fin	paires torsadées	fibre optique
Longueur max d'un segment	500m	200m	100m	2km
Nombre max de stations/segment	100	30	1024	1024

Remarque : Il existe aussi des classes Ethernet 1Mbit/s, etc. → pas présentées dans ce cours

3. Ethernet 802.3

Les éléments actifs Ethernet: permettent d'interconnecter tous les postes de travail d'un réseau local

	Rôles
Convertisseur (transceiver)	<ul style="list-style-type: none">permettre de se connecter au support de transmission (connecteur)permettre d'interconnecter deux types de supports différents
Carte réseau (chaque ordi)	<ul style="list-style-type: none">connecteur RJ45 pour câble à paires torsadéesconnecteur BNC pour câble coaxialconnecteur ST/SC pour fibre optique
Concentrateur (Hub)	<ul style="list-style-type: none">Répéteur multiportRenvoyer les données arrivant sur un port entrée/sortie (diffusion sur bus logique)Ré-amplifier les signaux en entrée
Commutateur (Switch)	<ul style="list-style-type: none">Comme un hubAnalyser l'adresse pour retransmettre uniquement vers le destinataire

3. Ethernet 802.3 – Evolutions

- **Fast Ethernet 802.3u** : débit jusqu'à 100Mbit/s
- **Gigabit Ethernet 802.3z**
- **10 Gigabit Ethernet 802.3ea (10-GBE)**

	802.3u		802.3z		802.3ae	
	100BaseT4	100BaseFX	1000BaseTX	1000BaseLX	10GBaseT	10GBaseER
Débit	100 Mbit/s	100 Mbit/s	1 Gbit/s	1 Gbit/s	10 Gbit/s	10 Gbit/s
Support (câble)	4 paires torsadées	fibre optique	paires torsadées	fibre optique $L = 10 \mu\text{m}$	paires torsadées	fibre optique $L = 1550\text{nm}$
Longueur maximale d'un segment	100m	2km-10km	100m	10km	100m	40km

Remarque : des évolutions vers le 25 Gbit/s et le 40 Gbit/s normalisées en 2015, 2016,...

3. Token Bus 802.4 et Token Ring 802.5

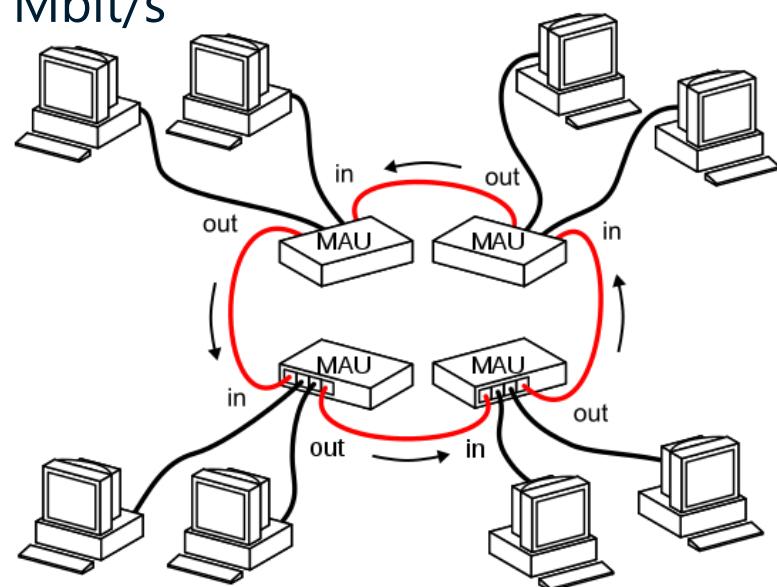
Description

- utilisent la méthode d'accès sans collision distribuée (jeton)
- définies en 1985 et très répandues durant une décennie mais quasiment disparues (très peu utilisées) au profit des architectures à haut/très haut débit
 - débit 802.4 : 1 Mbit/s, 5 Mbit/s, 10 Mbit/s
 - débit 802.5 : 1 Mbit/s à 4 Mbit/s, 1 Mbit/s à 16 Mbit/s

3. Token Bus 802.4 et Token Ring 802.5

Description

- utilisent la méthode d'accès sans collision distribuée (jeton)
- définies en 1985 et très répandues durant une décennie mais quasiment disparues (très peu utilisées) au profit des architectures à haut/très haut débit
 - débit 802.4 : 1 Mbit/s, 5 Mbit/s, 10 Mbit/s
 - débit 802.5 : 1 Mbit/s à 4 Mbit/s, 1 Mbit/s à 16 Mbit/s
- **MAU** (Médium Access Unit) : similaire à un Hub pour connecter des stations pour constituer un anneau principal



3. Wifi (Wireless Fidelity) 802.11

- **Assez semblable à Ethernet 802.3**

- usages: interconnexion au niveau LAN
- format: trames avec @MAC source et destination
- accès au support CSMA, mais avec CA et non CD

- **Transmissions radioélectriques** : très utile lorsque l'installation des câbles est impossible

- **Differentes bandes de fréquences :**

- 802.11, 802.11b, 802.11g dans la bande 2,4 GHz
- 802.11a et 802.11n dans la bande 5GHz

3. Wifi (Wireless Fidelity) 802.11

- **Modes d'exploitation**

- infrastructure : nécessite un **point d'accès au Wifi**
- poste à poste (ad hoc)

- **Débit et distance limitée selon des normes**

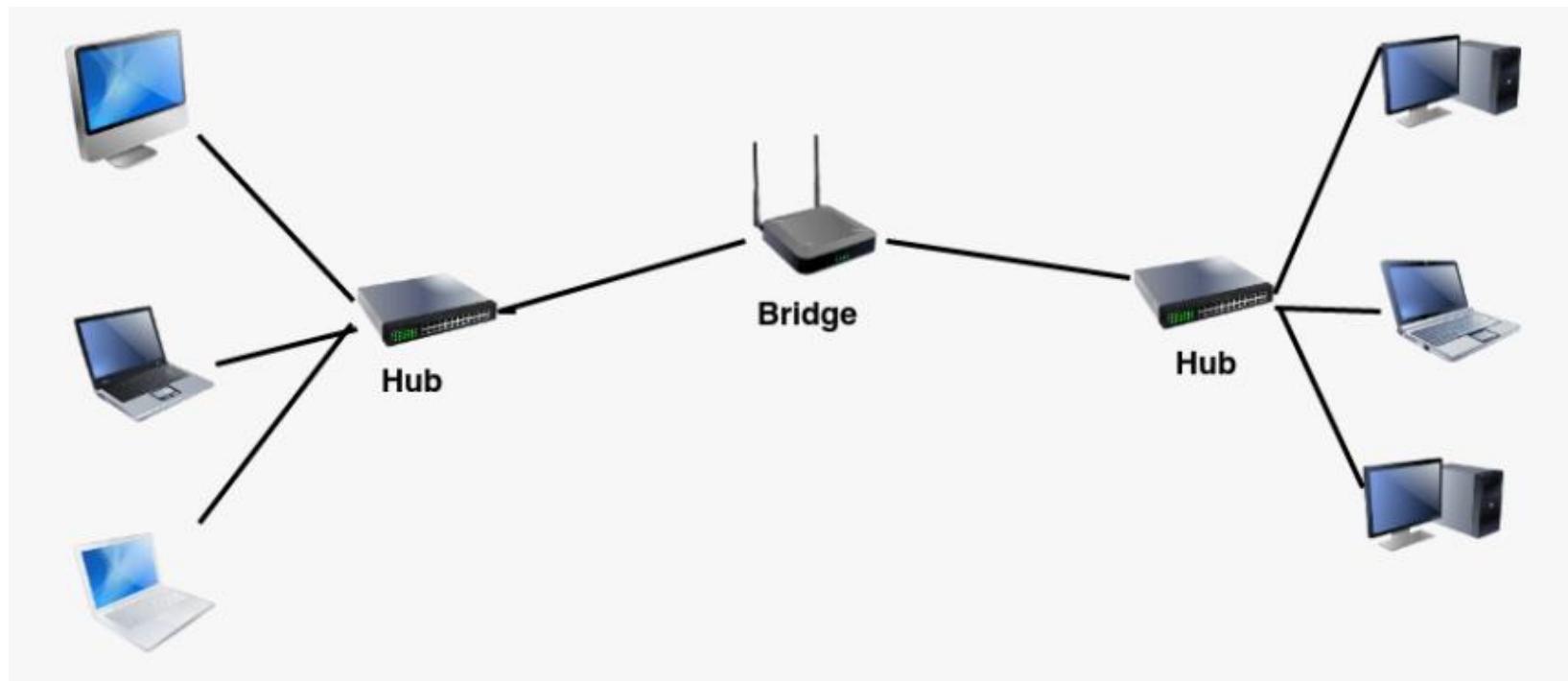
- 802.11a : 54 Mbit/s, 30 m
- 802.11b: 11 Mbit/s, 200 m
- 802.11g: 54 Mbit/s, 200 m
- 802.11ac : permet de transmission à haut débit (1.3 Gbit/s)

4. Les ponts

4. Les ponts

■ Principe

- élément permettant d'interconnecter des LANs dont des architectures répondant à des normes différentes
- convertir des trames au format nécessaire

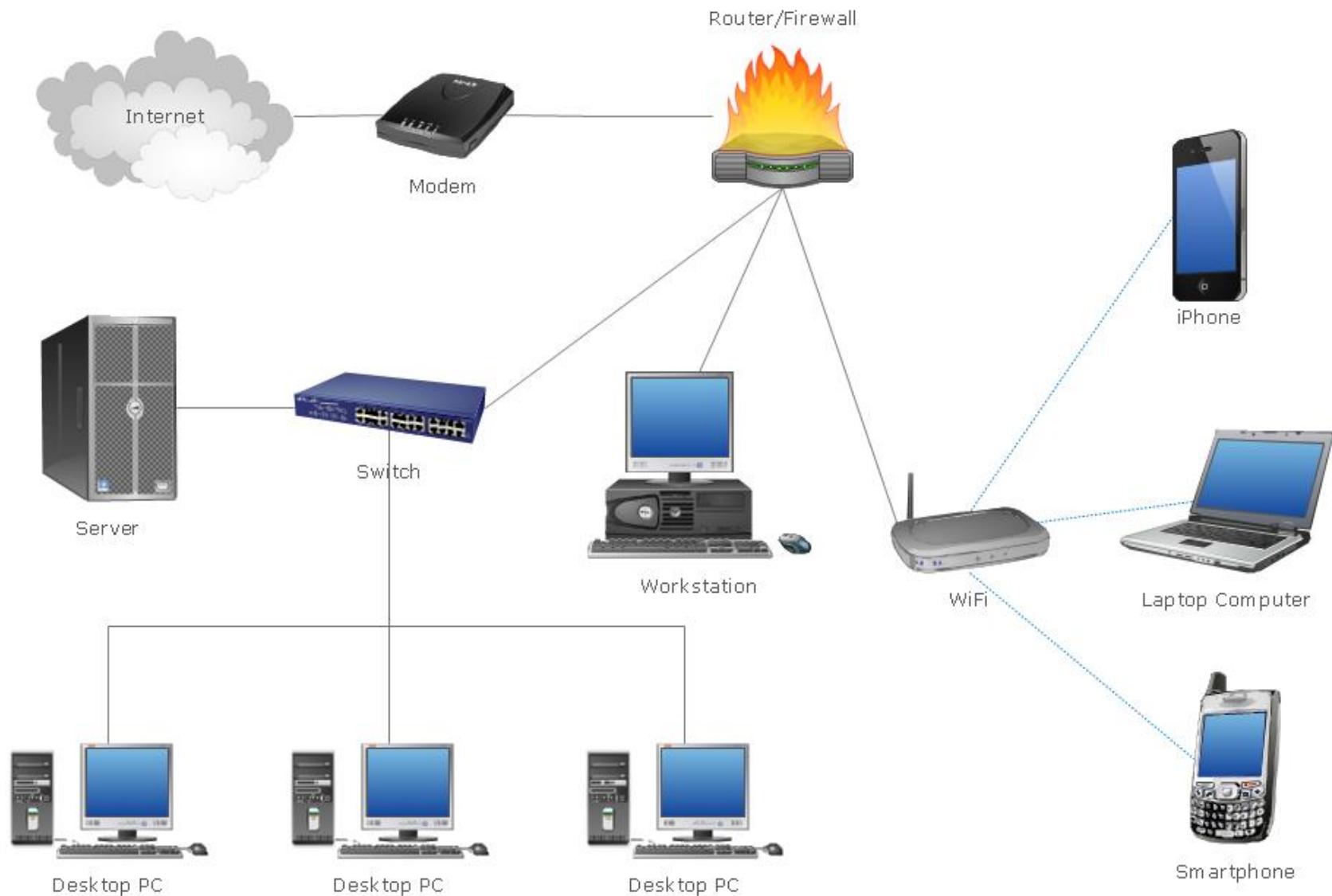


<https://www.pngitem.com/>

4. Les ponts

- **Pare-feu (Firewall)** : un pont particulier permettant de mettre en place de la sécurité par filtrage de données entre deux réseaux (normes identiques ou non)
- Se compose de trois éléments :
 - deux routeurs (reliés aux deux réseaux à interconnecter) → peuvent être remplacer par 2 cartes de réseaux
 - une passerelle applicative (*Proxy*) qui joue le rôle de liaison entre 2 interfaces de réseaux
- Essentiel pour connecter un réseaux local à Internet via un routeur

4. Exemple d'un diagramme des réseaux



<https://www.conceptdraw.com/>

– R2.04 – Communication et fonctionnement bas niveau –

Cours 5: La couche réseau

Minh Tan PHAM

BUT INFO, 2022-2023

IUT de Vannes, Université Bretagne Sud

minh-tan.pham@univ-ubs.fr



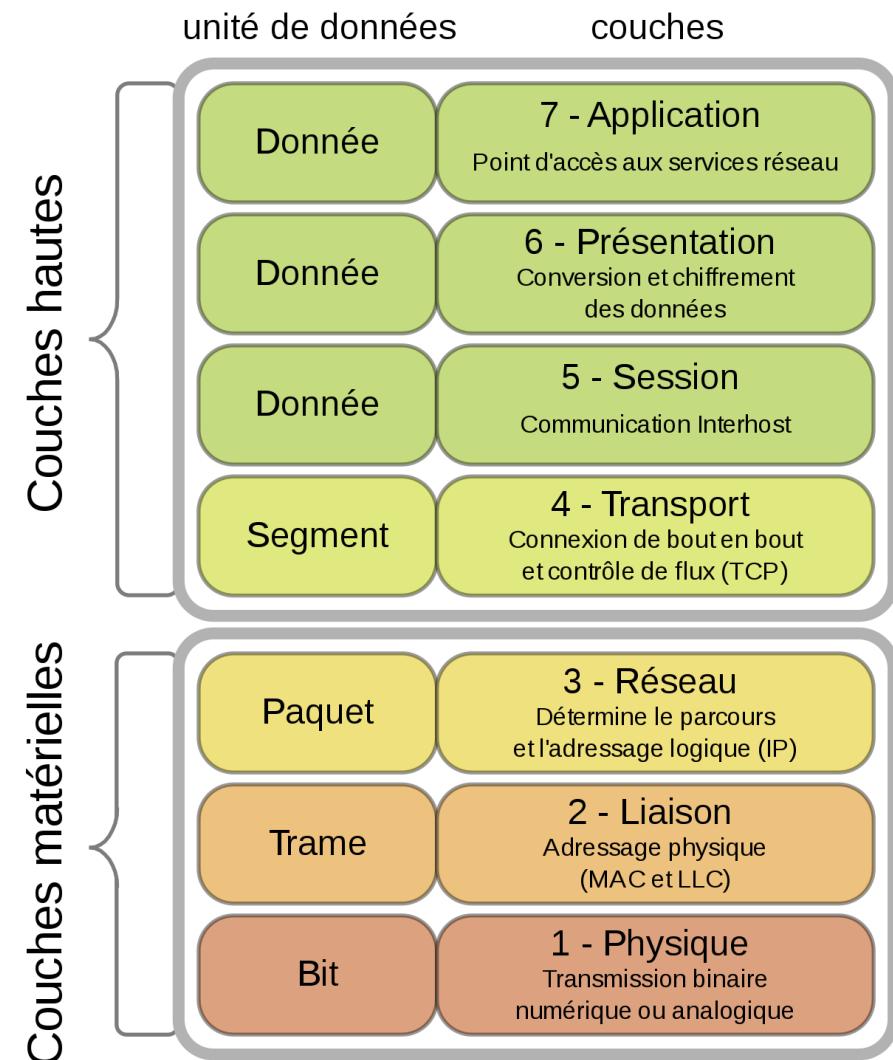
Plan du cours

- 1. Introduction**
- 2. Types de services**
- 3. Routage**
- 4. Les techniques de routage**
- 5. Le contrôle de congestion**

1. Introduction

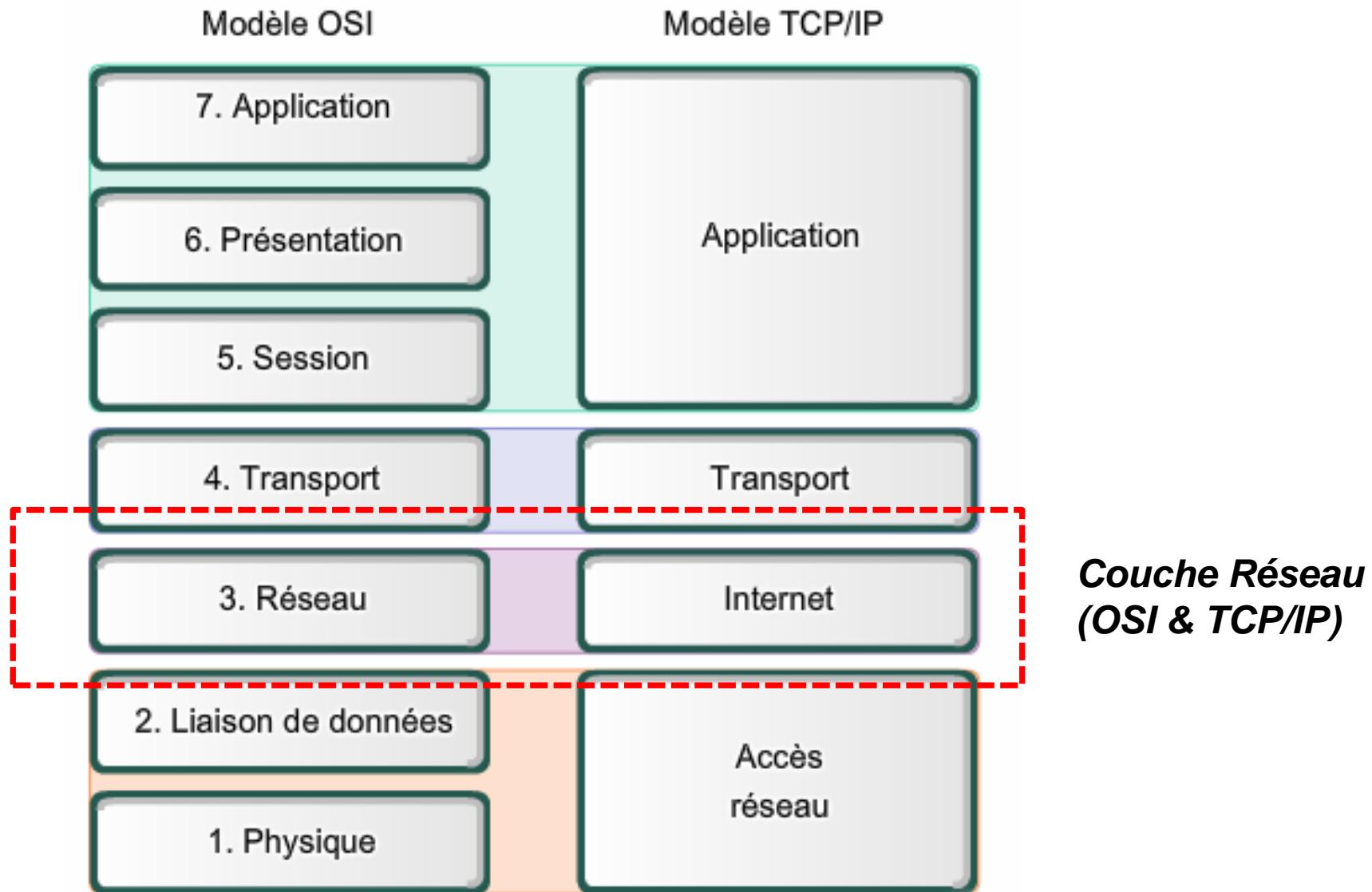
Couche réseau : Détermine le parcours des données et l'adressage logique (*routage et adressage des paquets*)

- Fournit les services à la couche transport
 - Acheminement des messages à travers le réseau
 - Contrôle de congestion (éviter des embouteillages) , et de la gestion de la qualité de service
 - Interconnexion de réseaux hétérogènes
- Connecte logiquement des hôtes qui ne sont pas directement reliés par un lien physique



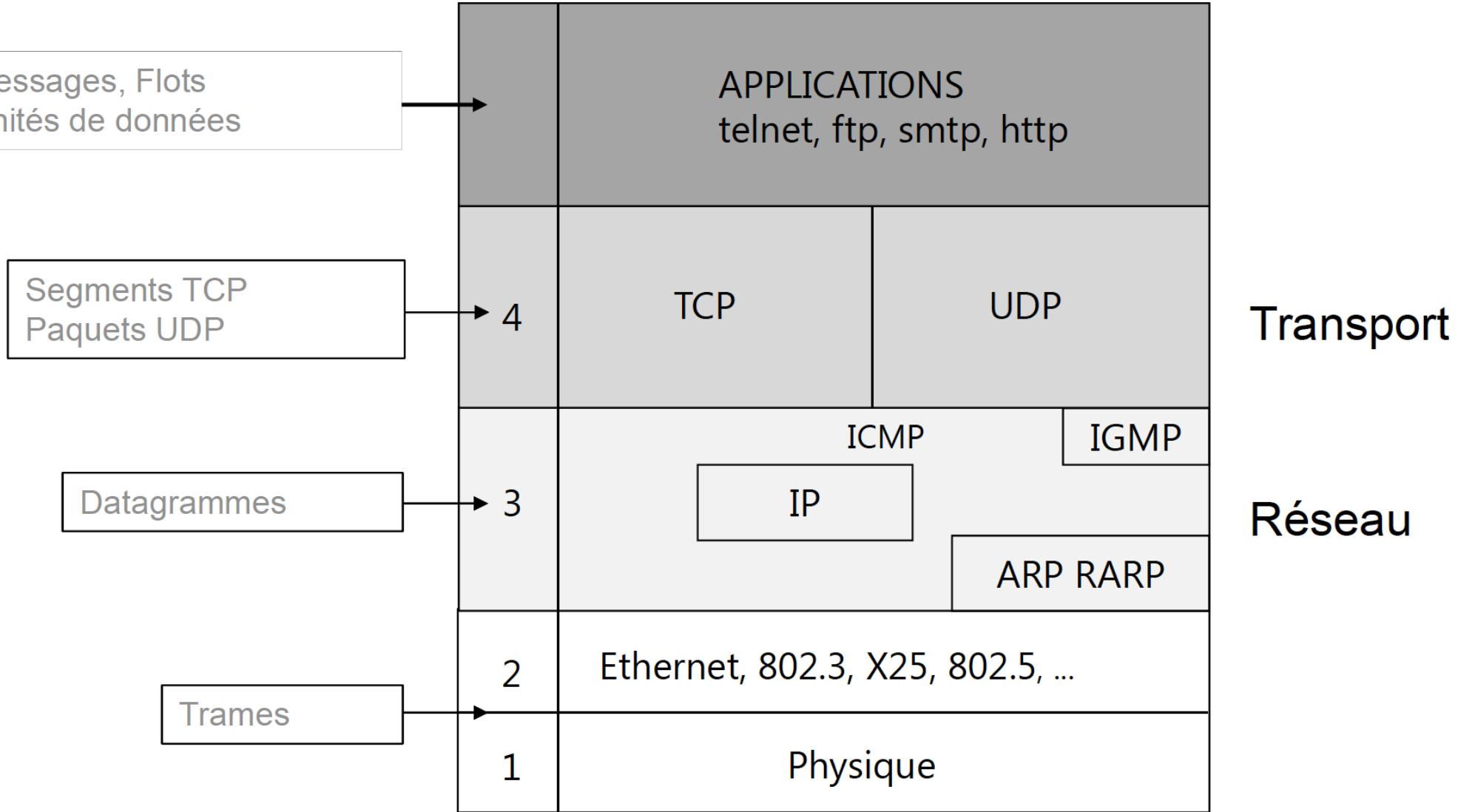
<https://fr.wikipedia.org/>

1. Introduction



<https://fr.wikibooks.org/>

1. Introduction

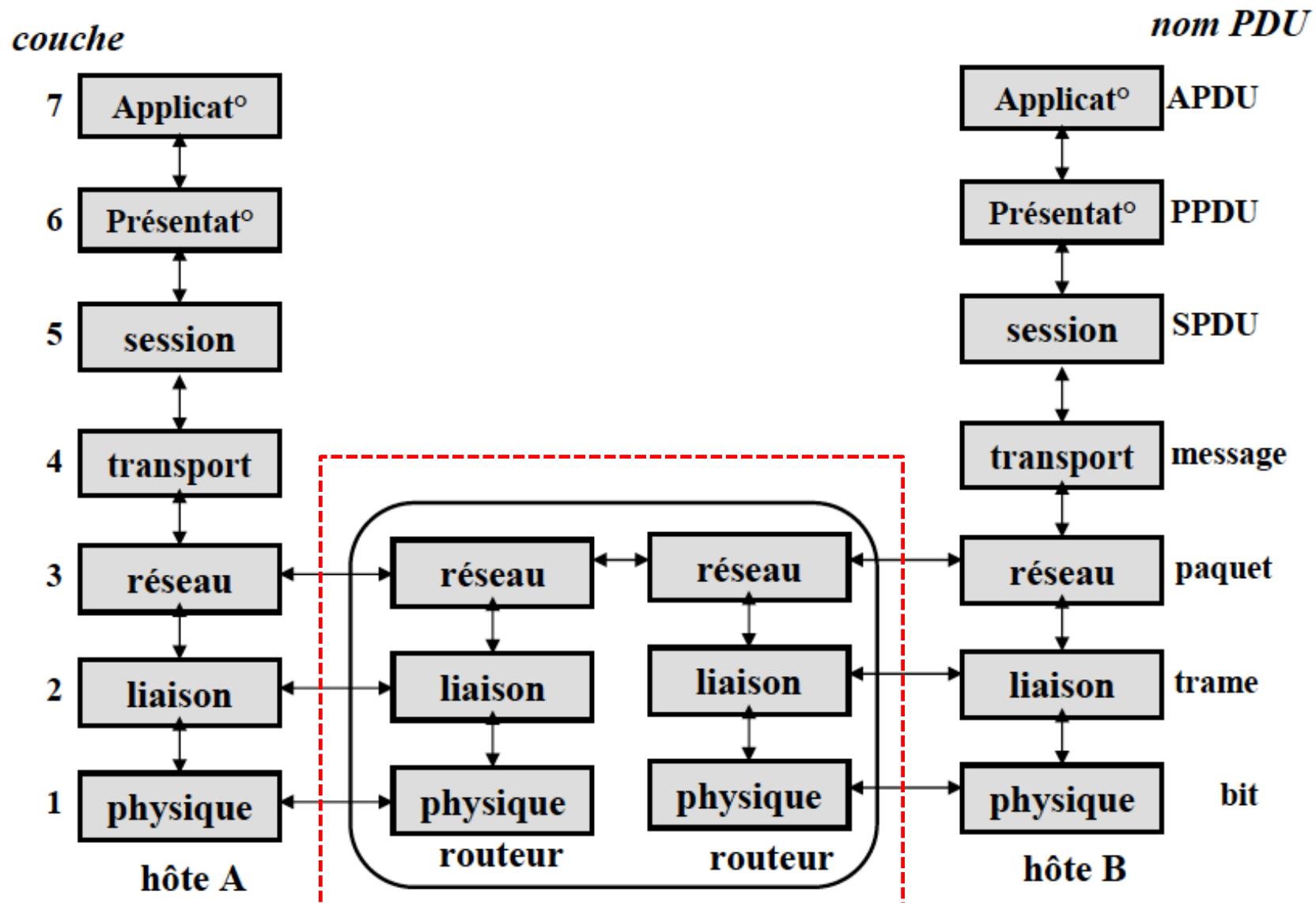


1. Introduction

Différents éléments d'interconnexion :

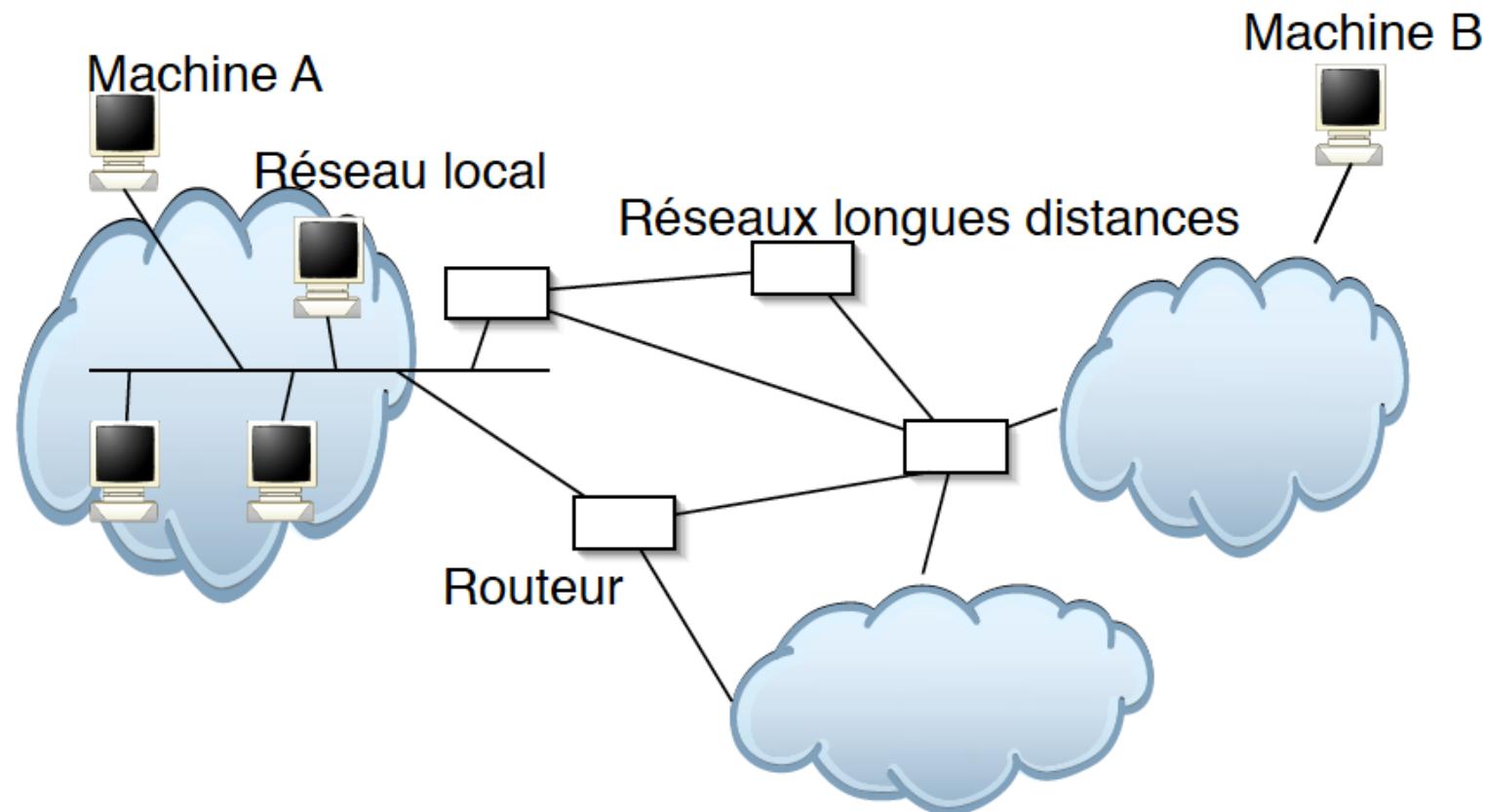
- **Répéteurs**
 - niveau physique, pas de conversion ni transcodage
 - répétition ou régénération des signaux, étendent de manière transparente le support physique
 - les réseaux sont identiques à partir de la couche liaison
 - rappel : hub et switch ont la fonction d'un répéteur
- **Ponts (bridges)**
 - niveau liaison de données, permet de passer d'un type de réseau à un autre
 - possibilité de filtrage
- **Routeurs (routers/gateways)**
 - niveau réseau
 - conversion de protocoles, incorporent des algorithmes de routage

1. Introduction



1. Introduction

Couche réseau : Détermine le parcours des données et l'adressage logique
(*routage et adressage des paquets*)



Routage : Déterminer la route des paquets de A vers B à travers le réseau ?

2. Types de services

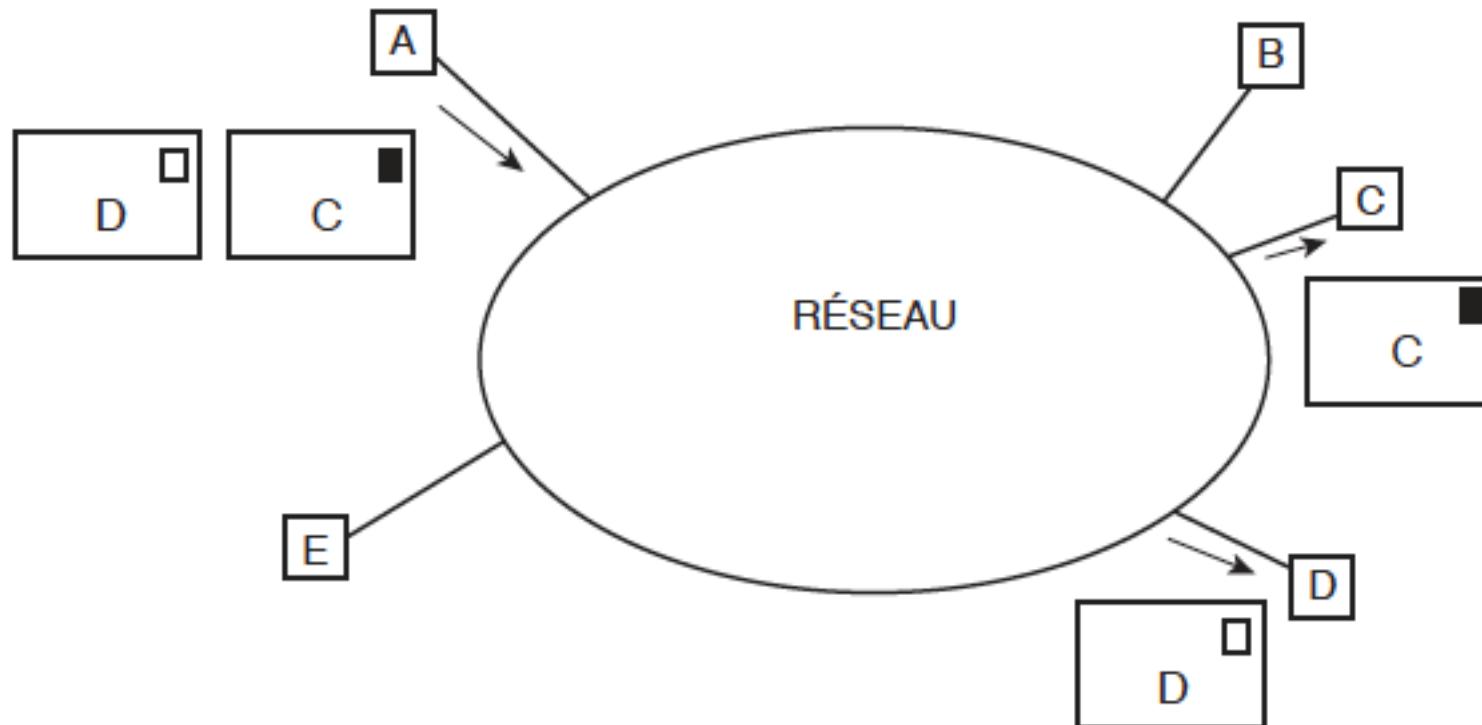
2. Types de services

- **Service sans connexion (non connecté):**
 - les paquets de données appelés **datagrammes** (incluent adresse complète émetteur et destinataire) sont acheminés indépendamment les uns des autres
 - Ex : IP (Internet Protocol)
- **Service orienté connexion (mode connecté):**
 - un chemin (ou **circuit virtuel**) est établi à la connexion, et libéré à la fin de la transmission.
 - lors d'une connexion, tous les paquets utiliseront un même chemin
 - Ex : X25.3, ATM (Asynchronous Transfert Mode)

2. Types de services

- **Service sans connexion (non connecté):**

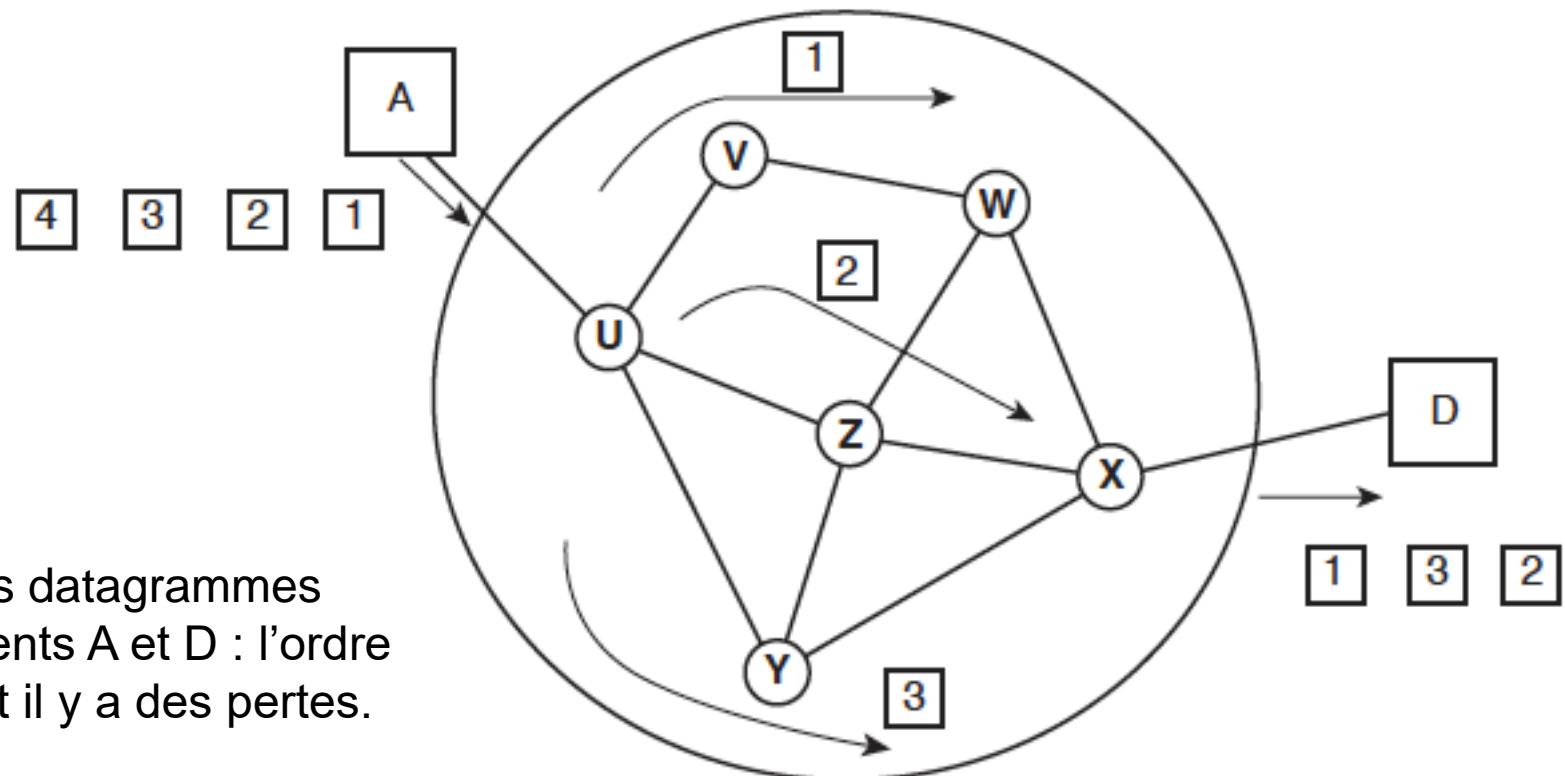
- les paquets de données appelés **datagrammes** (incluent adresse complète émetteur et destinataire) sont acheminés indépendamment les uns des autres
- Ex : **IP** (Internet Protocol)



2. Types de services

- **Service sans connexion (non connecté):**

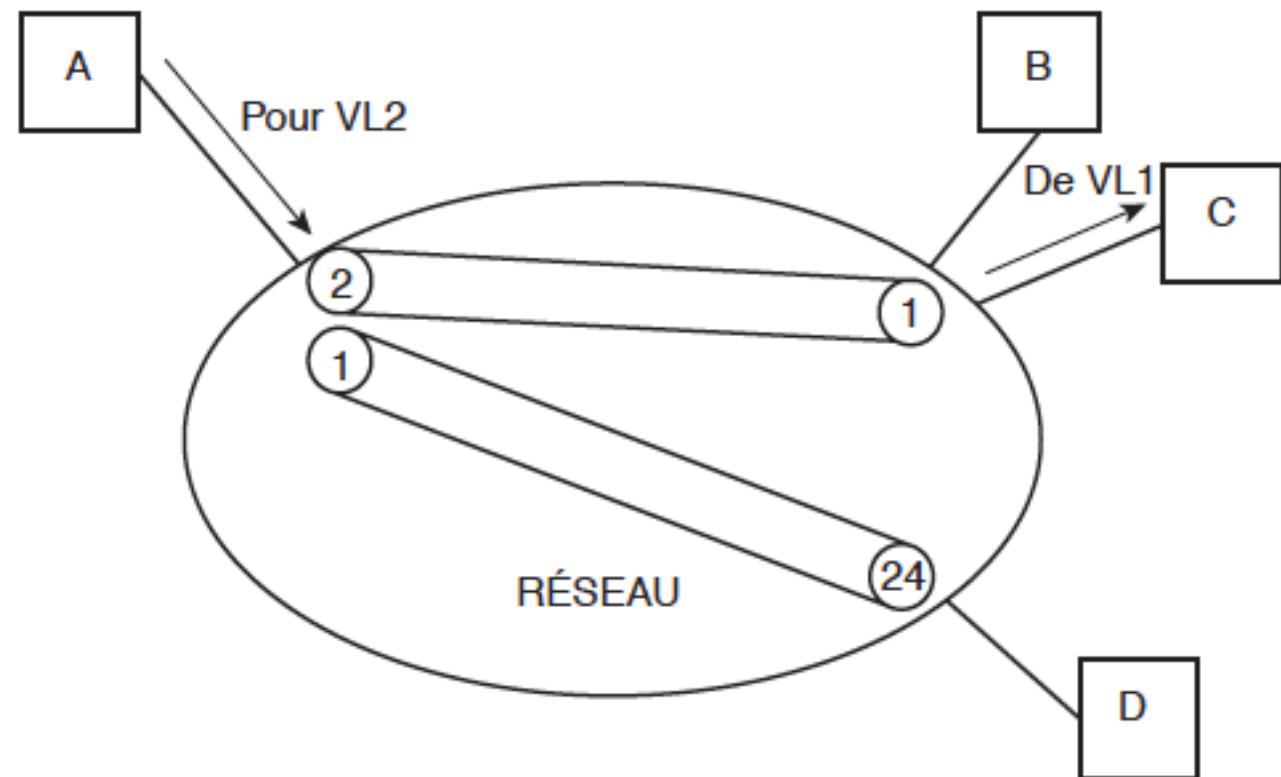
- les paquets de données appelés **datagrammes** (incluent adresse complète émetteur et destinataire) sont acheminés indépendamment les uns des autres
- Ex : **IP** (Internet Protocol)



Acheminement des datagrammes entre les équipements A et D : l'ordre n'est pas garanti et il y a des pertes.

2. Types de services

- **Service orienté connexion (mode connecté):**
 - un chemin (**circuit virtuel**) est établi à la connexion, libéré à la fin transmission
 - lors d'une connexion, tous les paquets utiliseront un même chemin
 - Ex : X25.3



2. Types de services

	Datagramme	Circuit virtuel
Phase établissement	Non nécessaire	Requise
Adressage	Adresses complètes (source et destination)	Numéro de circuit
Routage	Chaque paquet est indépendant	La même pour tous les paquets
Impact d'une panne d'un routeur	Aucun (sauf paquets perdus)	Tous les CV de ce routeur sont supprimés
Qualité de service	Difficile à garantir	Facile (relativement)
Contrôle de congestion	Difficile	Facile (relativement)

3. Routage

3. Routage

- **Généralités** : Chaque nœud du réseau a un certain nombre de canaux en entrée et en sortie. Le routage est la partie du logiciel qui décide sur quel canal de sortie doit être expédié un paquet qui arrive.
- **Deux cas :**
 - *En mode orienté connexion (CV)* : la décision de routage n'est prise qu'au cours de la phase d'établissement de la connexion.
 - *En mode non connecté (datagramme)* : chaque paquet est routé séparément → le routage est très important !
- **Objectifs :**
 - Exactitude, simplicité, justice (équité vis à vis des différents utilisateurs), optimisation, souplesse, et stabilité

3. Classification des algorithmes de routage

- **Centralisés, décentralisés**
 - **Centralisés** : les chemins sont calculés par un nœud particulier
 - **Décentralisé** : chaque nœud calcule les chemins
- **Statique ou dynamique**
 - **Statique** : les chemins sont fixes. Il faut les changer “à la main” en cas de modifications du réseau.
 - **Dynamique** : le choix des chemins s’adapte plus ou moins rapidement à des pannes réseaux et machines. L’adaptation à la charge des routeurs est très délicate et non implémentée.
- **Objectifs privilégiés par le routage**
 - plus court chemin
 - minimisation du temps moyen d’attente global

3. Classification suivant les informations dispo

- **Local ou global:** Les infos prise en compte pour calculer le chemin sont :
 - **globales** : connaissance de l'ensemble du réseau
 - **locales** : connaissance partielle du réseau
- **Nombres d'informations prises en compte**
 - La plupart des algorithmes associent un coût à un chemin (on parle de métrique)
 - Ce coût peut faire intervenir plus ou moins de paramètres : temps de propagation, débit, charge moyenne mesurée etc.

3. Notion de table de routage

- Chaque routeur possède une table de routage avec :
 - ensemble des destinations
 - une route à suivre (avec un coût associé)

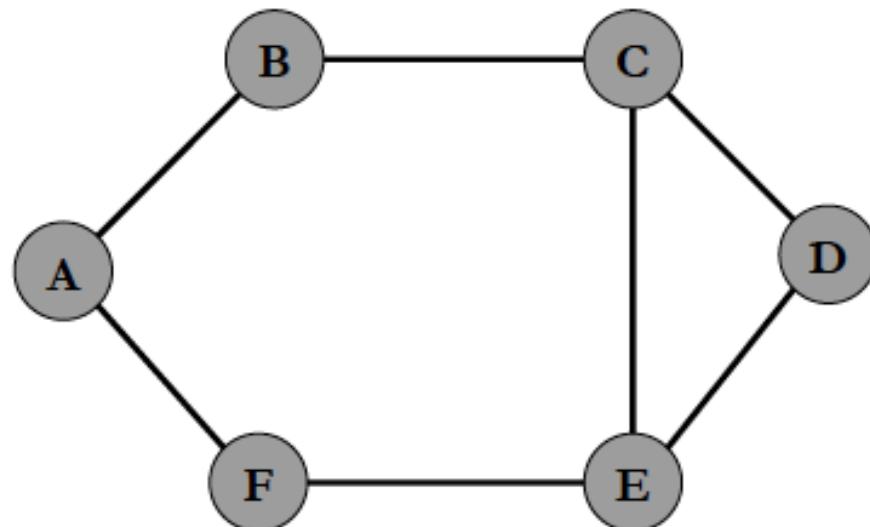


Table de routage de C

Dest.	Lien
A	CB
B	CB
D	CD
E	CE
F	CE

TD Routage !!

4. Techniques de routage

4. Techniques de routage

- **Algorithmes :**

- Routage par inondation (flooding)
- Routage du plus court chemin
- Routage par vecteur de distance
- Routage par informations d'état de lien
- Routage hiérarchique

4. Routage du plus court chemin

- Hypothèse :
 - Chaque nœud connaît la topologie exacte du réseau global
 - Chaque liaison se voit affecter un coût qui peut très différent selon qu'elle est vue d'une extrémité ou de l'autre.
 - La longueur d'un chemin est la somme des coûts des différentes liaisons
- Il existe plusieurs manières de compter la longueur d'un chemin : nombre de routeurs traversés, mesurer la distance géographique, la trafic sur un chemin...
- Il existe plusieurs algorithmes pour calculer le plus court chemin

TD Routage !!

4. Routage du plus court chemin

Réseau = Graphe $G(N, A, w)$

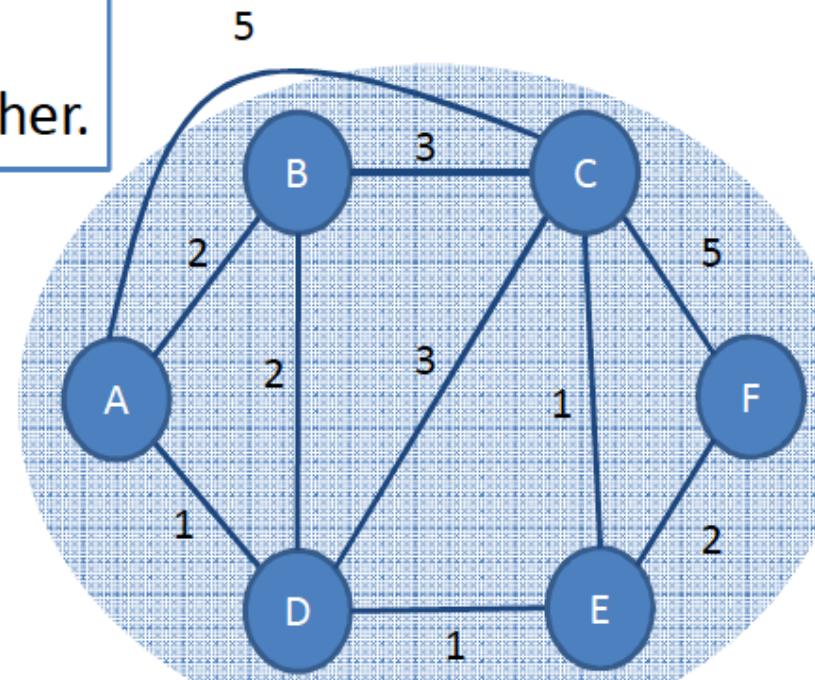
N = ens. nœuds : les routeurs

A = ens. d'arcs : les liaisons entre les routeurs

w = pondération des arcs : chaque chemin "coûte" plus ou moins cher.

Objectif —

Trouver le *bon chemin*
= **trouver le chemin de
coût minimal**



4. Routage du plus court chemin

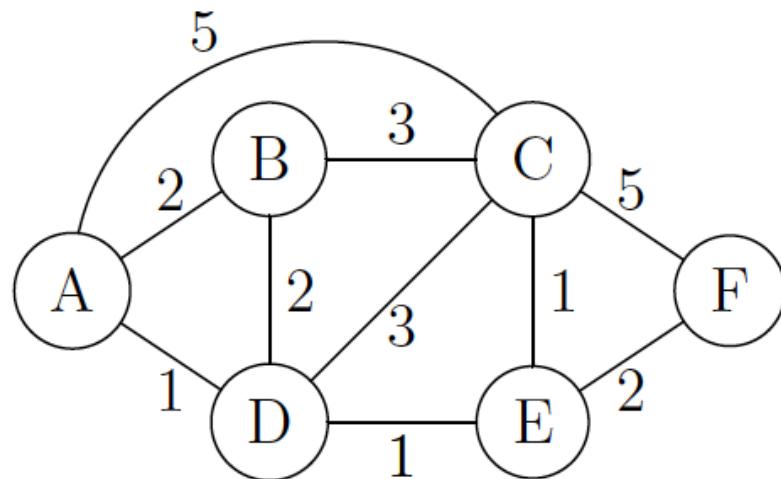
Algorithme de Dijkstra

On cherche à construire la distance du sommet s à tous les sommets du graphe (V, E) .

- Structures de données :
Ensemble de sommets marqués : M ; distance à s pour chaque sommet ($d(i)$); prédecesseur de chaque sommet ($p(i)$).
- Initialisation :
 $M = \{s\}$; $d(s) = 0$; $d(j) = d_{sj}$ pour tout $j \in N(s)$, ∞ sinon;
 $p(j) = s \forall j \in N(s)$.
- étape 1 : mise à jour des étiquettes
trouver $i \notin M$ tq. $d(i) = \min_{j \notin M}(d(j))$; $M \leftarrow M \cup \{i\}$.
Si $M = V$ on s'arrête.
- étape 2 : Mise à jour des distances
 $\forall j \in N(i)$ tq. $j \notin M$:
Si $d(j) > \min_{k \in N(j)}(d(k) + d_{kj})$ Alors :
 $p(j) \leftarrow \inf_{k \in N(j)}(d(k) + d_{kj})$; $d(j) \leftarrow \min_{k \in N(j)}(d(k) + d_{kj})$; fin si
retour à l'étape 1.

4. Routage du plus court chemin

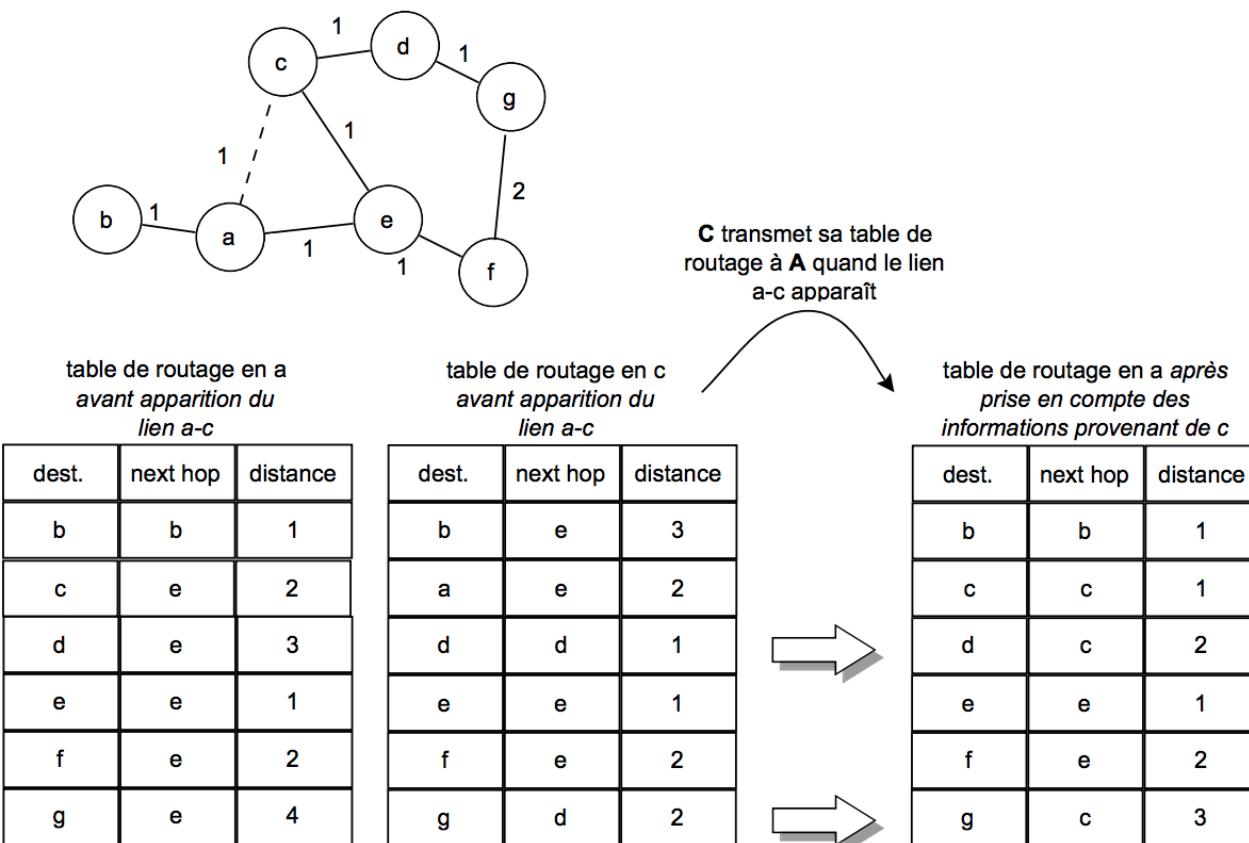
Algorithme de Dijkstra



	M	$d(B)$	$d(C)$	$d(D)$	$d(E)$	$d(F)$
Init	A	2	5	1	∞	∞
1	A, D	2	4	1	2	∞
2	A, D, E	2	3	1	2	4
3	A, D, E, B	2	3	1	2	4
4	A, D, E, B, C	2	3	1	2	4
5	A, D, E, B, C, F	2	3	1	2	4

4. Routage à vecteur de distance

- Exemples : RIPv1 ; RIPv2 (masques variables)
- Structure de données minimale : destination / next hop / distance
- Exemple de mise à jour :
le routeur transmet les couples (destination, distance)



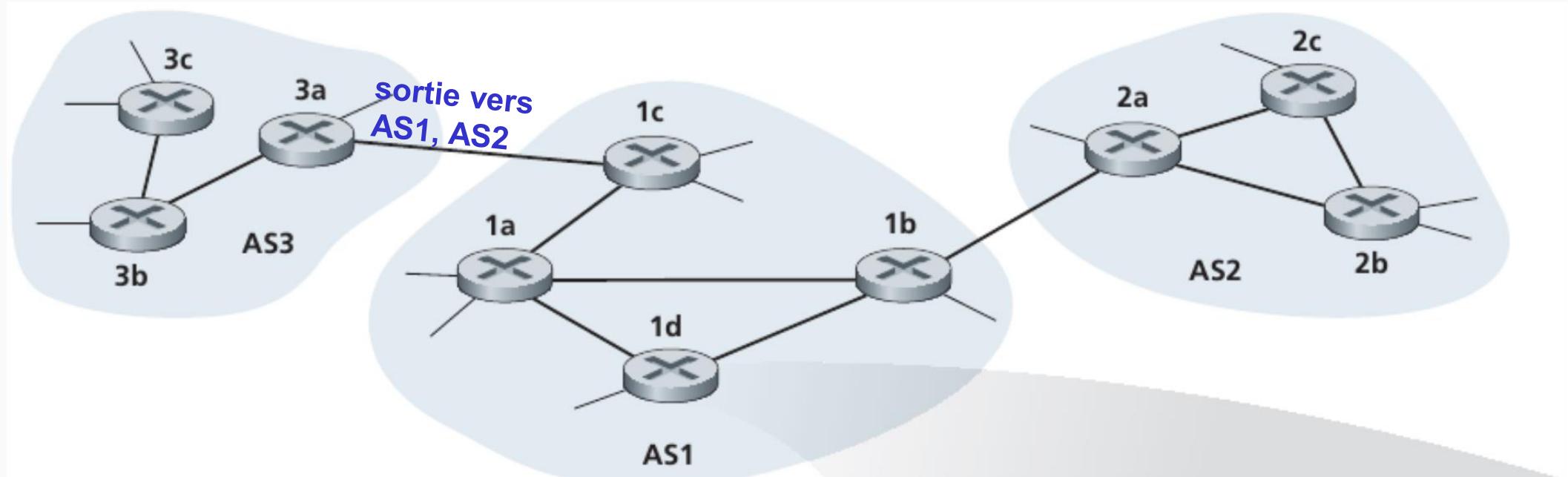
TD Routage !!

4. Routage hiérarchique

- **Problème :**
 - grand réseau → taille des tables de routage importante !
- **Solution :**
 - découper/diviser le réseau en plusieurs régions ou sous-réseaux
 - chaque routeur va alors posséder dans sa table trois infos:
 - les ports de sortie pour accéder à chaque destinataire dans sa région
 - les ports de sortie permettant d'accéder à chacune des autres régions du réseau
 - un port de sortie à utiliser par défaut pour une adresse de destinataire inconnue
 - routage hiérarchique est basé sur la technique de routage à vecteur de distance

4. Routage hiérarchique

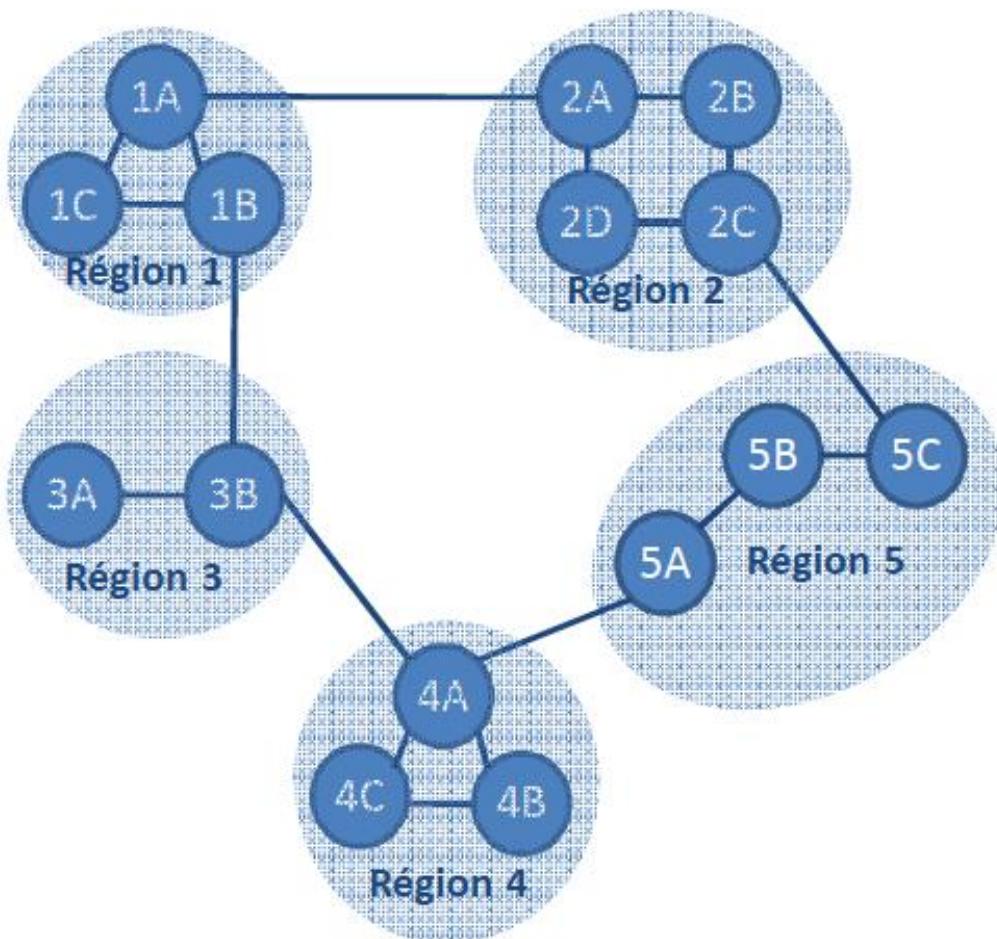
- Illustration :



TD Routage !!

4. Routage hiérarchique

■ Illustration :



Dest	Next Hop	Sauts (hops)
1A	1A	1
1B	<i>1B</i>	1
1C	-	0
2A	1A	2
2B	1A	3
2C	1A	4
2D	1A	3
3A	<i>1B</i>	3
3B	<i>1B</i>	2
4A	<i>1B</i>	3
4B	<i>1B</i>	4
4C	<i>1B</i>	4
5A	<i>1B</i>	4
5B	<i>1B</i>	5
5C	1A	5

Dest	Next Hop	Sauts (hops)
1A	1A	1
1B	<i>1B</i>	1
1C	-	0
2	1A	2
3	<i>1B</i>	2
4	<i>1B</i>	3
5	<i>1B</i>	4

Moins d'espace !!!

Technique utilisée pour le routage Internet (Routage IP → prochain cours)

– R2.04 – Communication et fonctionnement bas niveau –

Cours 6: La pile TCP/IP

Minh Tan PHAM

BUT INFO, 2022-2023

IUT de Vannes, Université Bretagne Sud

minh-tan.pham@univ-ubs.fr



Plan du cours

- 1. Introduction**
- 2. Protocole IP**
- 3. Adressage IP**
- 4. Routage IP**
- 5. Protocole ARP, ICMP**
- 6. Couche Transport : UDP et TCP**
- 7. Service Internet : DNS, FTP, Telnet, SSH**

1. Introduction

- Modèle développé par l'agence ARPA (Advanced Research Project Agency)
 - 1969 : **ARPAnet** : favorise les réseaux résistants aux destructions partielles avec grand succès
 - 1978 : stade opérationnel
 - 1983 : les protocoles TCP/IP deviennent des standards militaires
 - 1990 : explosion d'IP en Europe → devenu un standard au niveau des réseaux locaux et plus particularité des réseaux étendus comme l'Internet
- Remarques :
 - Modèle TCP/IP → une simplification du modèle OSI plus pragmatique et représentatif des technologies existantes
 - Il est important de comprendre les mécanismes et les protocoles qui permettent une intégration entre les couches TCP/IP et celles OSI
 - TCP/IP correspond à **deux notions** :
 - la notion de modèle basé sur des couches (modèle TCP/IP)
 - la notion d'implémentation : appellation aux logiciels basés sur les protocoles TCP/IP

1. Introduction

- **Couche Accès réseau :** accéder au support physique pour la transmission de données
 - Acheminement des données sur la liaison
 - Coordination de la transmission de données (synchronisation)
 - Format des données
 - Conversion des signaux (analogique/numérique)
 - Contrôle des erreurs à l'arrivée
 - ...
- **Couche Internet :** couche la plus importante
 - Le protocole IP (*Internet Protocol*)
 - Le protocole ARP (*Address Resolution Protocol*)
 - Le protocole ICMP (*Internet Control Message Protocol*)
 - Le protocole RARP (*Reverse Address Resolution Protocol*)
 - Le protocole IGMP (*Internet Group Management Protocol*)

1. Introduction

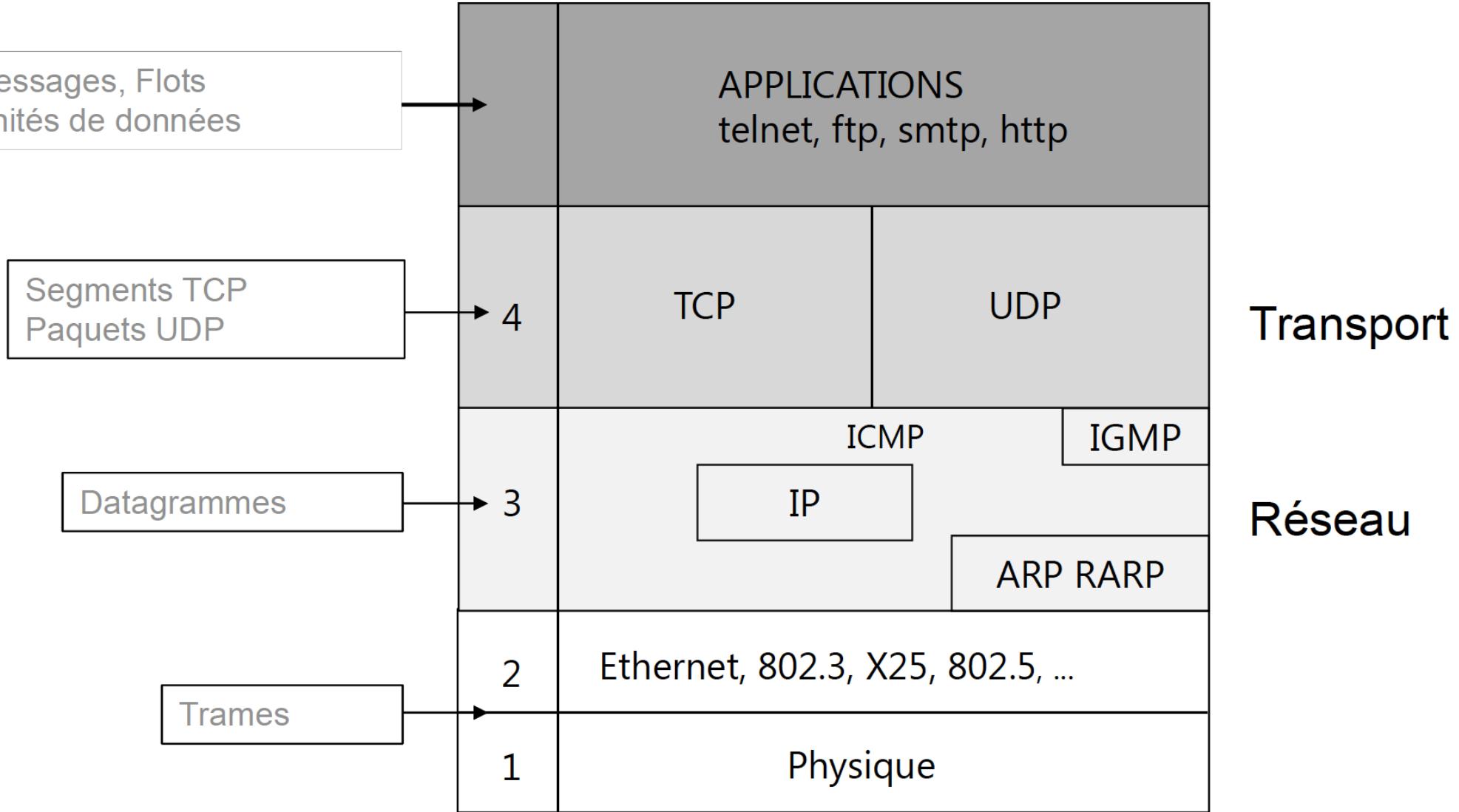
- **Couche Transport** : permettre à des entités paires de soutenir une conversation.

Officiellement deux protocoles :

- *TCP (Transmission Control Protocol)* : protocole orienté connexion qui assure le contrôle des erreurs
- *UDP (User Datagram Protocol)* : protocole non orienté connexion dont le contrôle d'erreurs est peu fiable

- **Couche Application** : contient des protocoles et des services de haut niveau
 - *SMTP (Simple Mail Transfer Protocol)*
 - *Telnet*
 - *HTTP (HyperText Transfer Protocol)*
 - *FTP (File Transfer Protocol)*

1. Introduction



2. Protocole IP

2. Protocole IP

Deux versions :

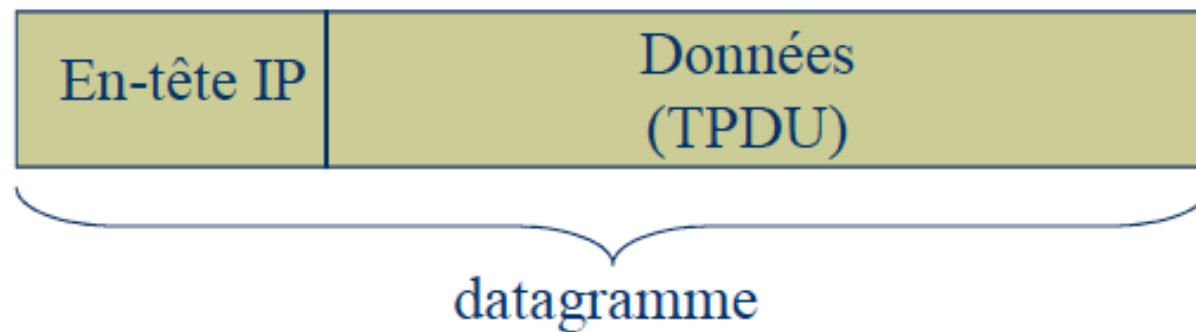
- **IPv4 :**
 - Version actuellement répandue
 - Codage des adresses sur 32 bits
- **IPv6 :**
 - Déploiement en cours, mais encore limité
 - 18.23% (jan 2018), 25.91% (jan 2019)
 - 24.6% (France), 41.36% (Allemagne), 34.5 (USA)
 - Codage des adresses sur 128 bits
 - Cohabite avec la version 4
 - Nécessite une nouvelle pile de protocoles
 - Implémenté sur la plupart des OS modernes

2. Fonctions 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)
- Assure le routage
- Peut fragmenter les messages

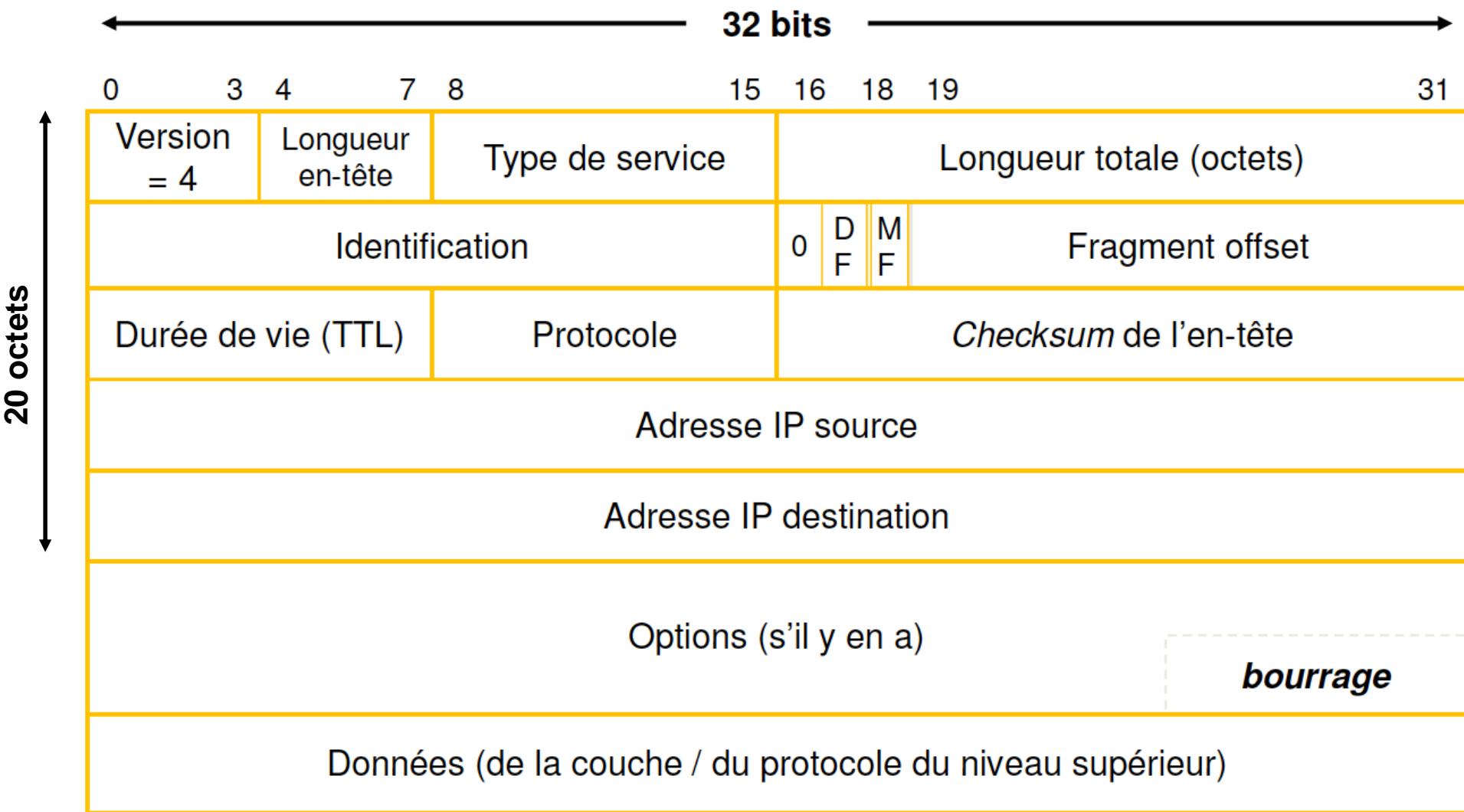
2. Format du paquet IP (v4)

- 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

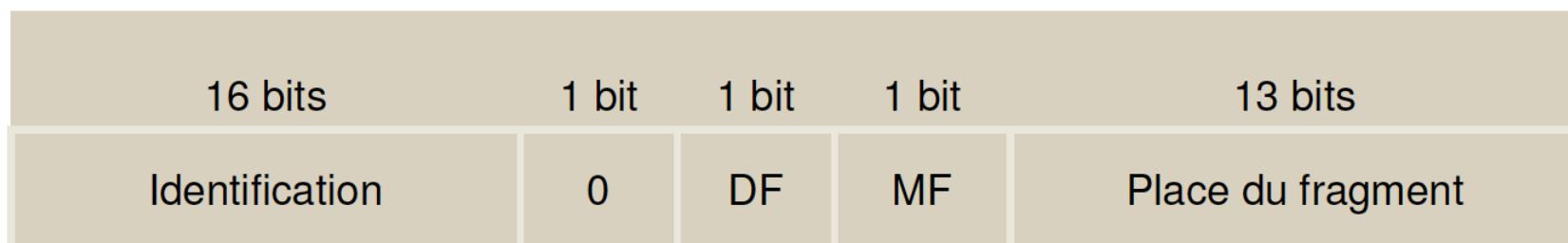
2. Format du paquet IP (v4)



2. Format du paquet IP (v4)

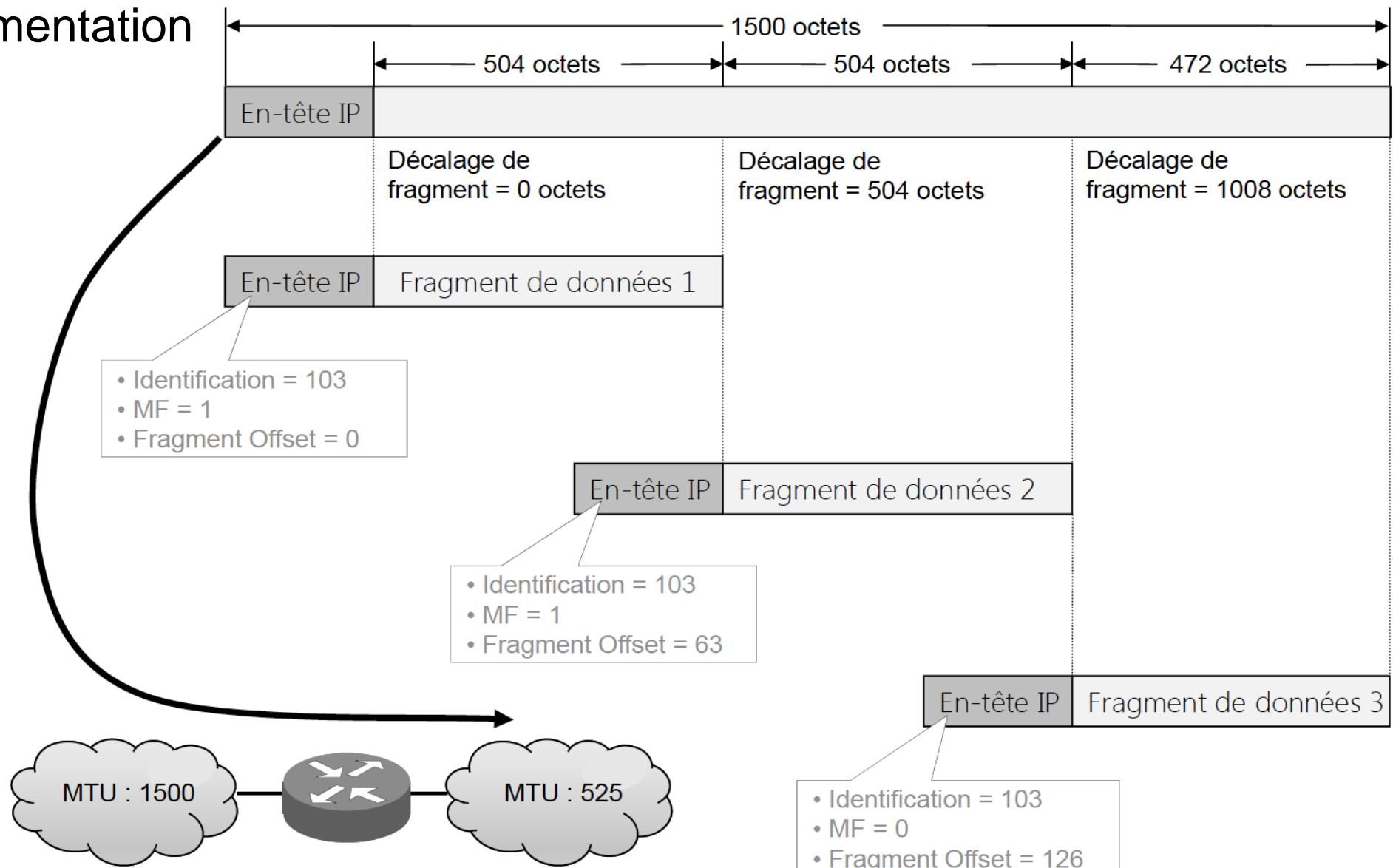
Fragmentation : champs de l'en-tête

- **Identification** : numér unique (pour l'émetteur)
 - Si le paquet est fragmenté après, tous les fragments le portent
- **Place du fragment** : position du 1^{er} octet du fragment dans le datagramme original (non fragmenté)
 - Découpe des fragments en multiples de 8 octets
- **DF (don't fragment) = 1** \Rightarrow le paquet ne doit pas être fragmenté
- **MF (more fragments)**
 - MF = 0 dernier fragment
- Drapeaux par défaut (paquet non fragmenté) : DF = MF = 0



2. Format du paquet IP (v4)

Fragmentation



3. Adressage IP (v4)

3. Structure de l'adresse IP (v4)

- Chaque interface réseau d'un appareil possède une adresse IP unique
- Codée sur **32 bits (4 octets)** en notation décimale pointée

11000000.10101000.00001010.10000010 = 192.168.10.130

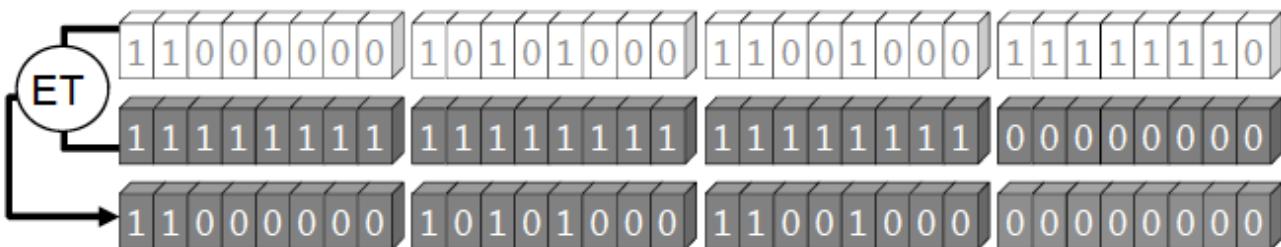
- Structurée en deux parties :
 - le préfixe, donnant le numéro de réseau : ID de réseau (**netId**)
 - le suffixe → numéro de la machine (hôte) dans ce réseau (**hostId**)
- Un masque (**netmask**) est associé à cette adresse

Interface : eth0

Adresse IP : 192.168.200.254

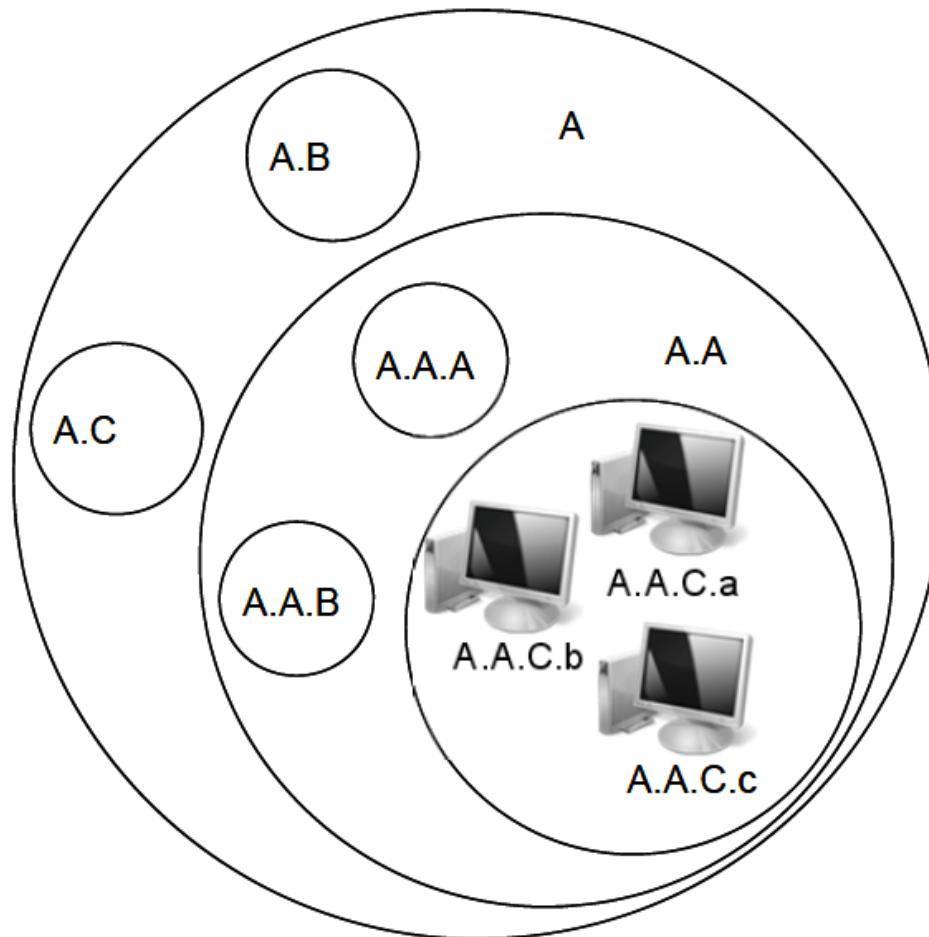
Masque réseau : 255.255.255.0

Préfixe réseau : 192.168.200.0



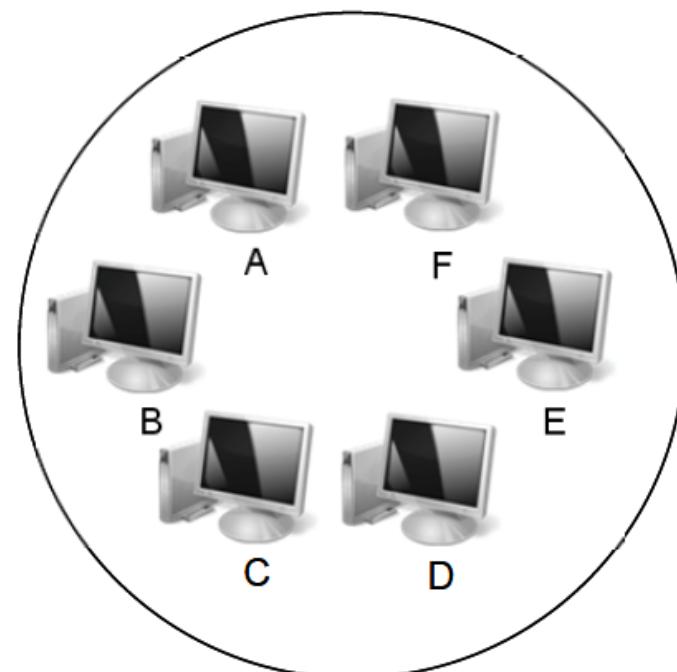
3. Espace d'adressage IP (v4)

Adresse IP



Espace d'adressage hiérarchique

Adresse Ethernet

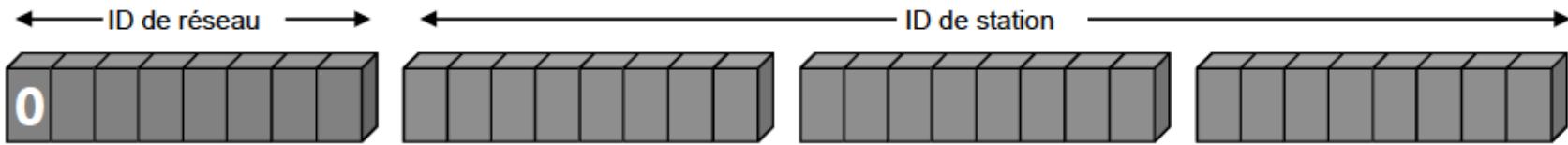


Espace d'adressage plat

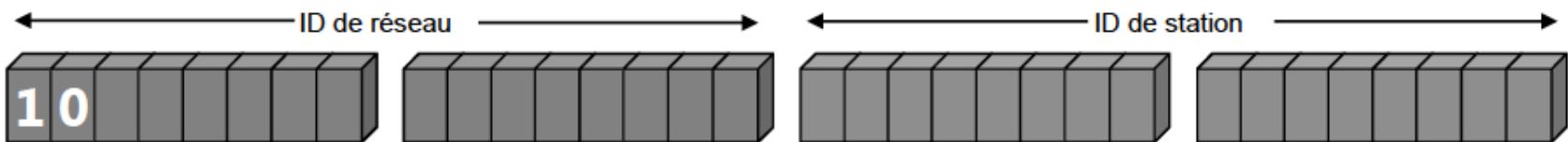
3. Notion de l'adresse IP (v4)

- Notion classique
 - Adresse IP/masque en décimal pointé
 - Ex : 192.168.200.254/255.255.255.0
- Notion condensée
 - Adresse IP/nombre de bits 1 du masque
 - Ex : 192.168.200.254/24

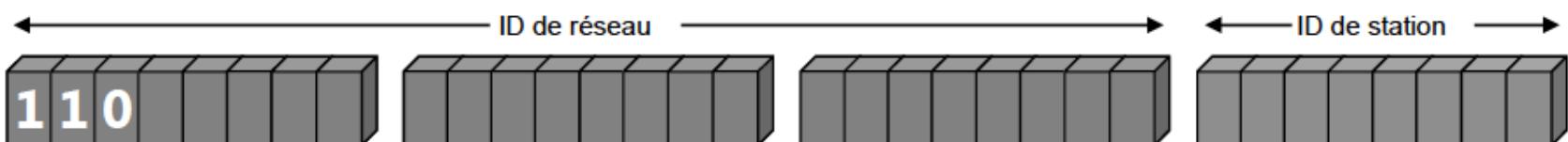
3. Classes des adresses IP



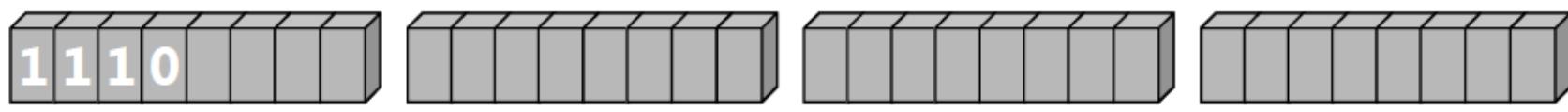
Classe A : de 0.0.0.0 à 127.255.255.255



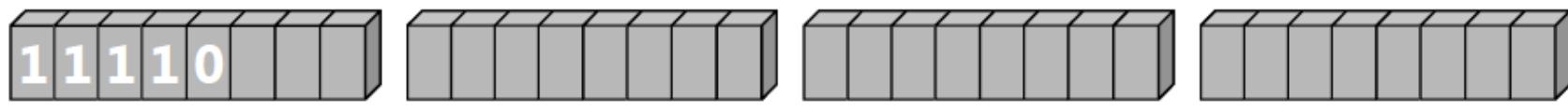
Classe B : de 128.0.0.0 à 191.255.255.255



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

3. Adresses IP spéciales

- **Adresse de réseau :**
 - Id réseau suivi par des 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 (broadcast) :**
 - Id réseau suivi par des 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

3. Adresses IP spéciales

- Adresse de **machine** ou **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**
 - désigne la machine locale (localhost)
- <netId=1> et <hostId=0> : masque du réseau
- 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

3. Sous-réseaux IP (subnet)

- Diviser un gros réseau unitaire en ce qui apparaît comme plusieurs sous-réseaux (**subnetting**)
 - permet meilleure structuration du réseau du site
 - décidé par l'administrateur du site
 - adresse de sous-réseaux prélevé sur la partie **hostId**
 - 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

3. Sous-réseaux IP (subnet)

Exemple : réseau de classe B 140.30.0.0

ID réseau 16 bits (140.30)	ID machine 16 bits
-------------------------------	-----------------------

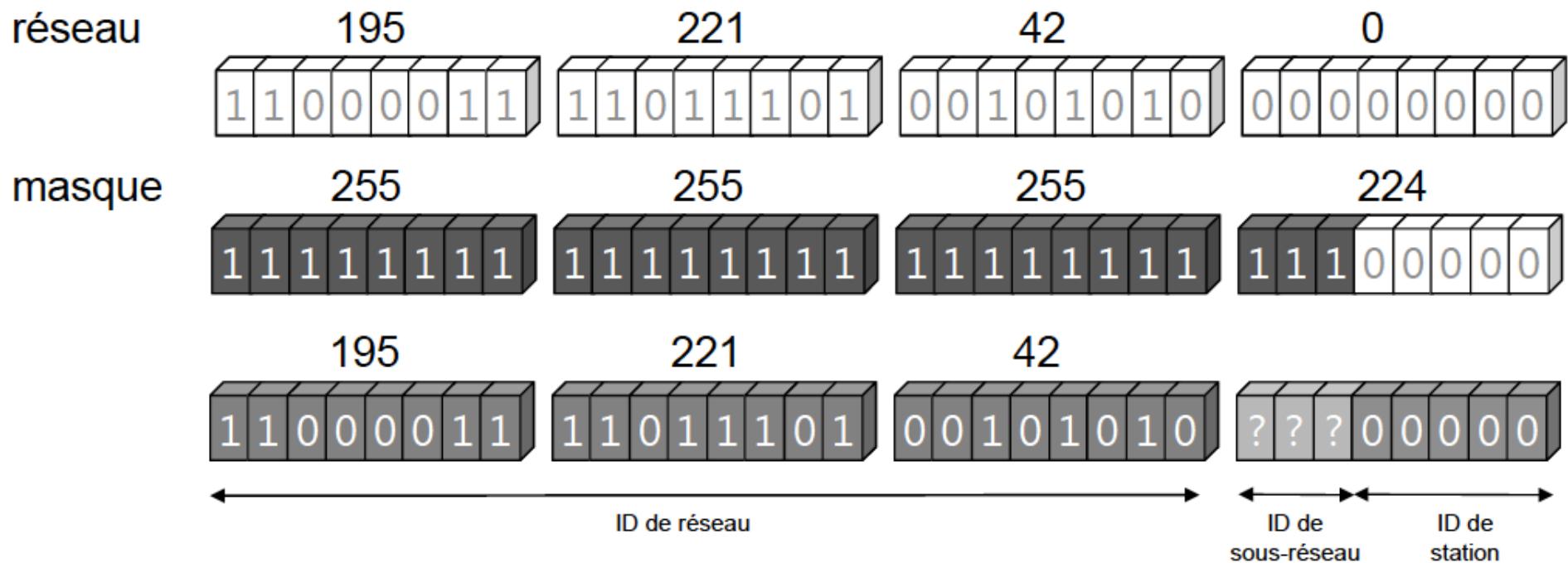
- masque de réseau par défaut 255.255.0.0 si aucun sous-réseau n'est défini

ID réseau 16 bits (140.30)	ID sous-réseau 8 bits	ID machine 8 bits
-------------------------------	--------------------------	----------------------

- masque 255.255.255.0 si présence de (au plus 254) sous-réseaux (de 254 hôtes chacun)

3. Sous-réseaux IP (subnet)

Exemple : réseau de classe C 195.221.42.0



3. Sous-réseaux IP (subnet)

Exemple 2: réseau, sous-réseau, masque, hôte

hôte	192	55	12	120
	11000000	00110111	00001100	01111000
masque	255	255	255	240
	11111111	11111111	11111111	11110000
n sous-réseau	0	0	0	112
	00000000	00000000	00000000	01110000
n d'hôte	0	0	0	8
	00000000	00000000	00000000	00001000
broadcast	192	55	12	127
	11000000	00110111	00001100	01111111

3. Sous-réseaux IP (subnet)

Exemple de création de sous-réseaux

- 130.79.153.28/23 : adresse réseau 130.79.152.0
- Adresse réseau : 10000010 01001111 10011000 00000000
- Si je veux créer 3 sous-réseaux : j'ai besoin de 2 bits supplémentaires dans l'adresse réseau pour les créer (je crée ainsi 4 sous-réseaux).
 1. 10000010 01001111 10011000 00000000 : 130.79.152.0/25
 2. 10000010 01001111 10011000 10000000 : 130.79.152.128/25
 3. 10000010 01001111 10011001 00000000 : 130.79.153.0/25
 4. 10000010 01001111 10011001 10000000 : 130.79.153.128/25
- Chaque sous-réseau pourra adresser $126 \text{ hôtes} = 2^7 - 2$

3. Sur-réseaux IP (supernet)

- Regroupement de plusieurs réseaux en un seul bloc
 - concept développé en 1985 pour optimiser l'espace d'adressage IP
 - interconnexion des sous-réseaux impérativement par des routeurs
 - 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)

3. Sur-réseaux IP (supernet)

- 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

↓
Plus petite
adresse du bloc
↓
Masque de sur-réseau
↓
Masque par défaut

11000000	00110111	00010000	00000000
11111111	11111111	11111000	00000000
11111111	11111111	11111111	00000000

← → *supernet mask*

← → *default netmask*

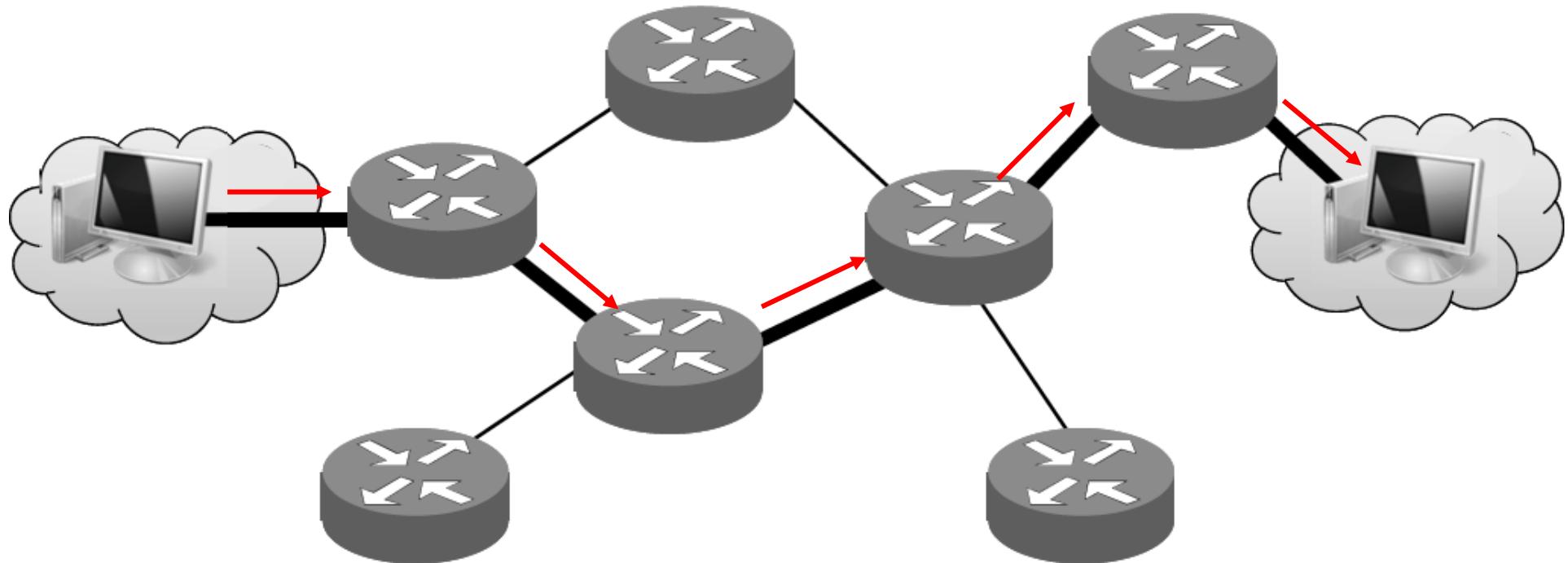
3. IPv6

- Successeur de la version actuelle d'IP (IPv4)
 - Maintient les meilleures fonctions d'IPv4
 - En ajoute de nouvelles quand elles sont nécessaires
- Simplification de l'en-tête des datagrammes
 - 7 champs (contre 14 pour IPv4)
 - permet aux routeurs de traiter les datagrammes plus rapidement et améliore globalement leur débit
- Principales améliorations: Adresses codées sur 16 octets contre 4 octets (128 bits contre 32)
 - offrant un espace d'adressage quasi illimité (3.4×10^{38} adresses)
 - notées sous forme de 8 groupes de 4 chiffres hexadécimaux séparés avec le symbole deux-points
 - Exemple : **2001:0db8:0000:85a3:0000:0000:ac1f:8001**

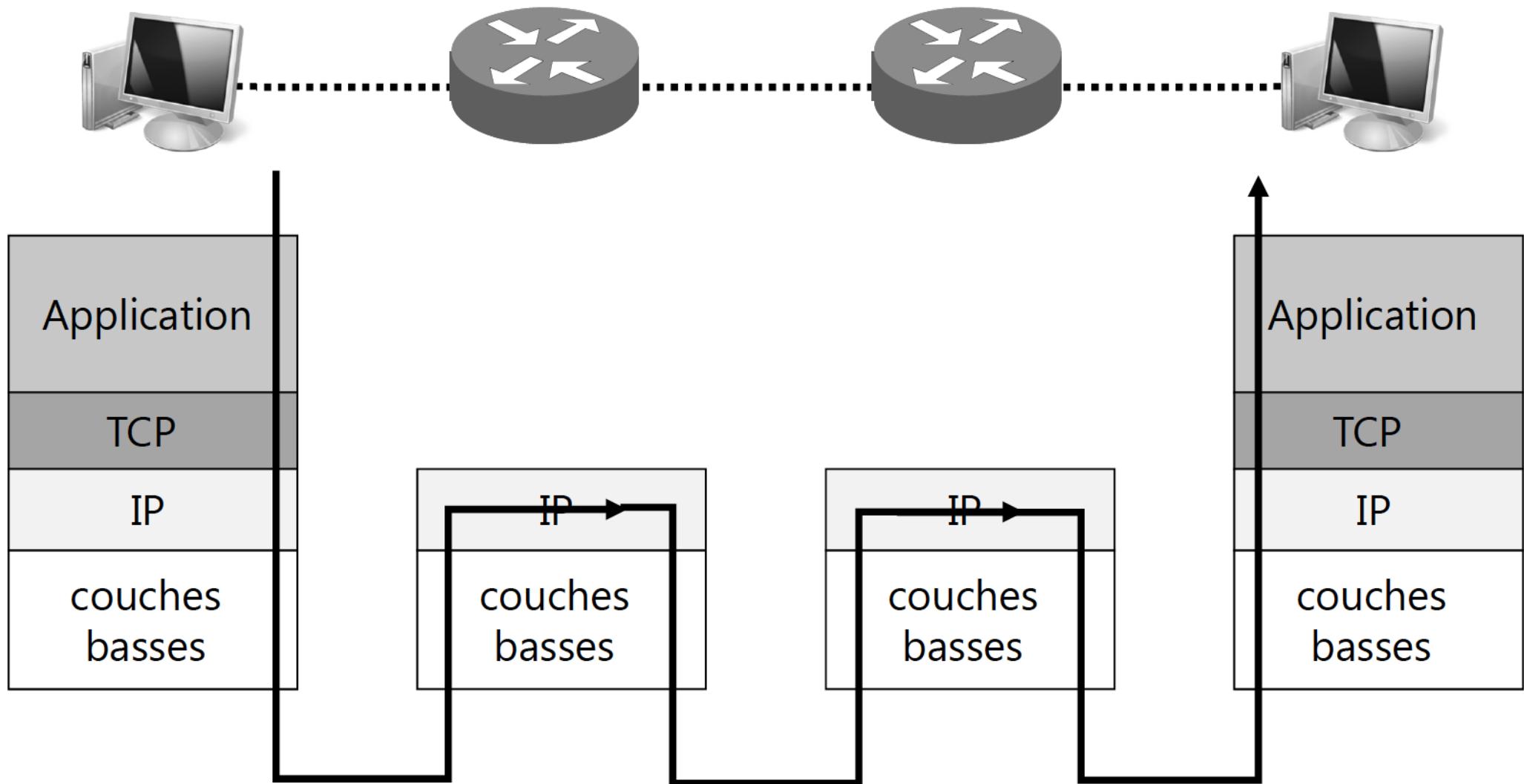
4. Routage IP

4. Routage IP - Principes

- 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
- 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



4. Routage IP



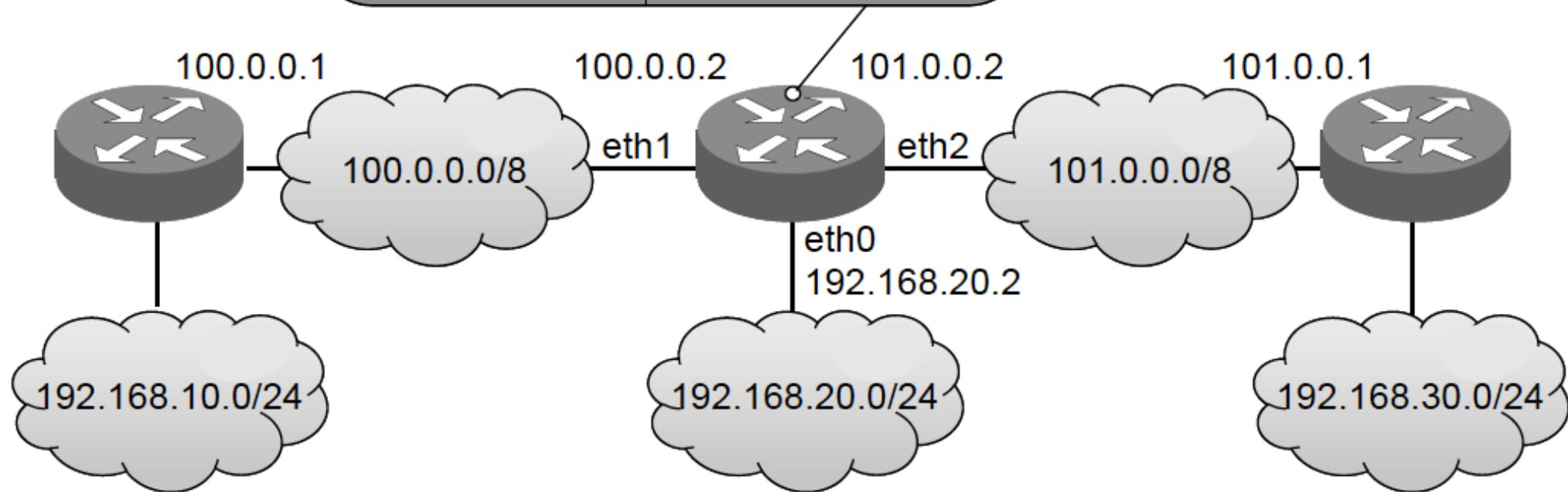
4. Routage IP - Principes

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 2 types d'informations
 - des adresses réseau des destinations
 - et le moyen d'atteindre
 - si le réseau est directement connecté à l'appareil → **nom d'interface**
 - sinon → **adresse du routeur** de prochain pas (**next hop**) situé sur la route vers ce réseau
- Protocole **RIP** (Routing Information Protocol) avec la technique *Next-Hop Routing*

4. Routage IP

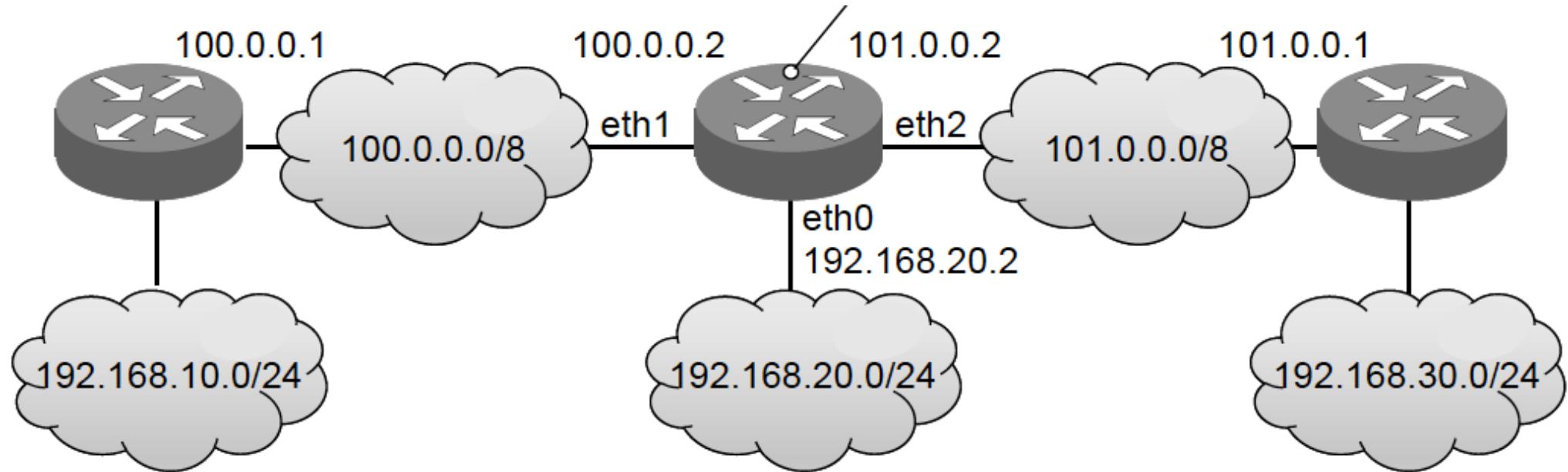
Destination	Moyen de l'atteindre
192.168.20.0/24	eth0
100.0.0.0/8	eth1
101.0.0.0/8	eth2
192.168.10.0/24	100.0.0.1
192.168.30.0/24	101.0.0.1



4. Routage IP

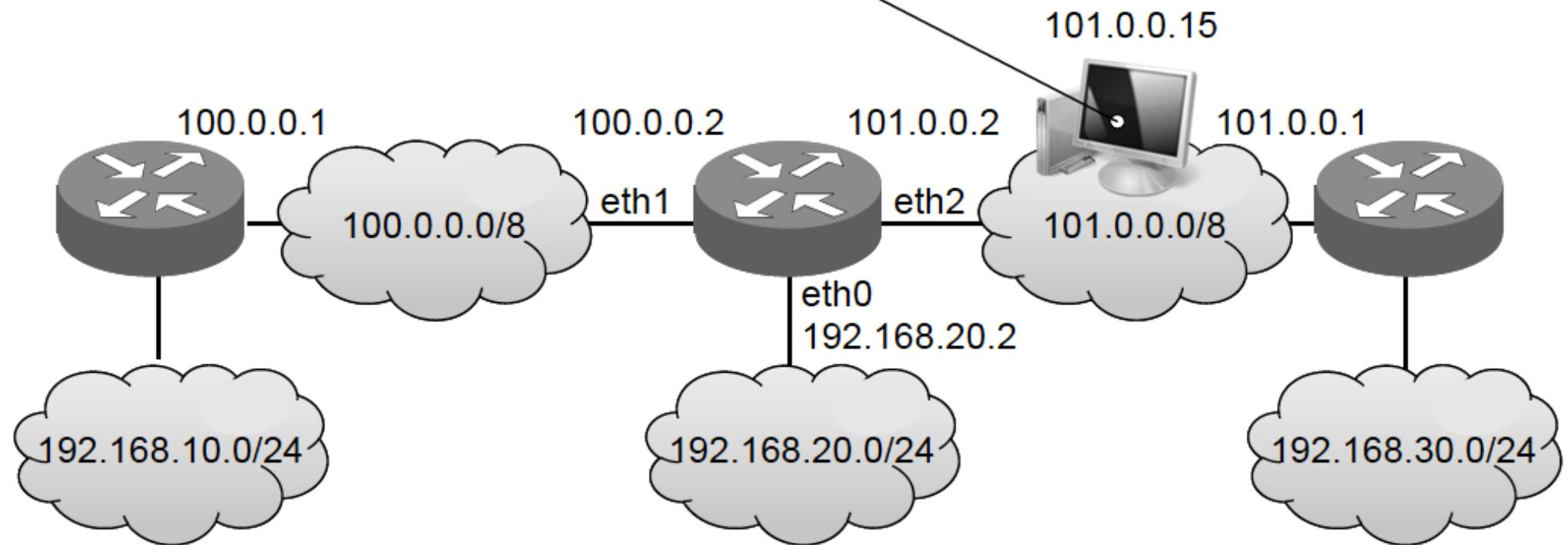
(routeur suivant)

Adresse dest.	netmask	passerelle	interface	vecteur dist.
192.168.20.0	255.255.255.0	192.168.20.2	192.168.20.2 (eth0)	0
100.0.0.0	255.0.0.0	100.0.0.2	100.0.0.2 (eth1)	0
101.0.0.0	255.0.0.0	101.0.0.2	101.0.0.2 (eth2)	0
192.168.10.0	255.255.255.0	100.0.0.1	100.0.0.2 (eth1)	1
192.168.30.0	255.255.255.0	101.0.0.1	101.0.0.2 (eth2)	1



4. Routage IP

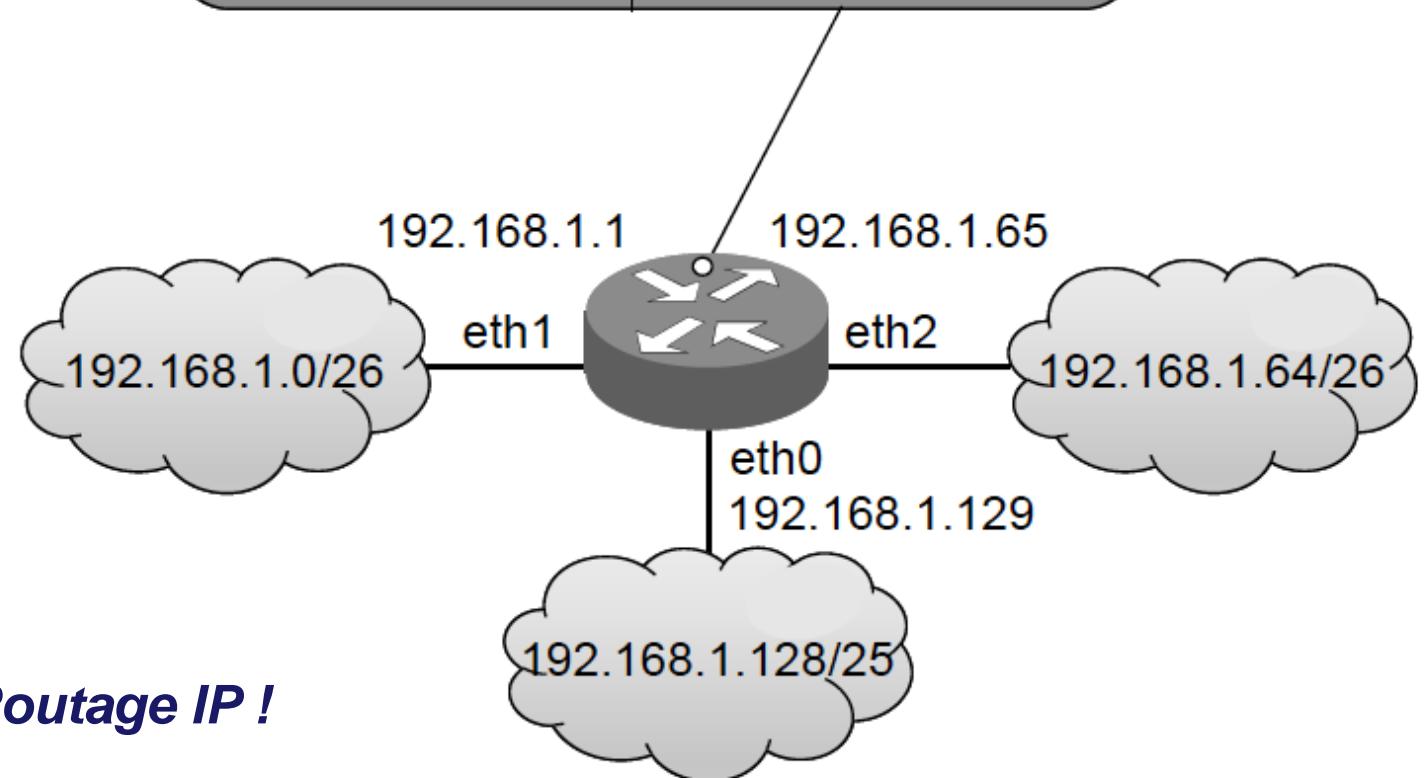
Destination	Moyen de l'atteindre
101.0.0.0/8	eth0
192.168.30.0/24	101.0.0.1
0.0.0.0	101.0.0.2



4. Routage IP

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



TD Adressage et Routage IP !

4. Routage IP - Algorithme

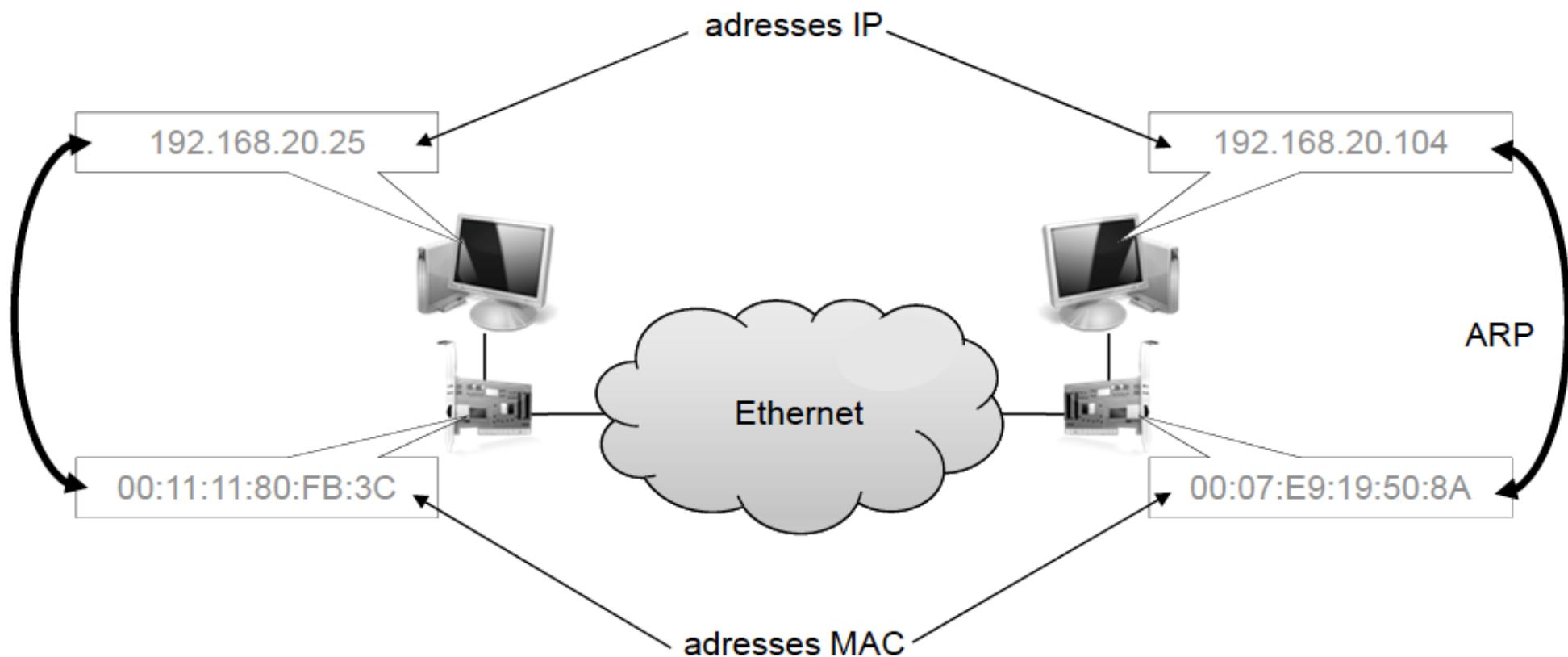
Algorithme exécuté lors de l'émission d'un paquet de données

1. calcul du préfixe de réseau de l'adresse destination à l'aide du masque
2. recherche du préfixe dans la table de routage :
 - si le préfixe correspond à celui d'un réseau directement connecté → **remise directe** du paquet sur réseau (protocole ARP) et fin de 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

5. Protocole ARP et ICMP

5. ARP (Address Resolution Protocol)

- Correspondance entre adresse IP (32 bits) et adresse physique au niveau MAC (48 bits)

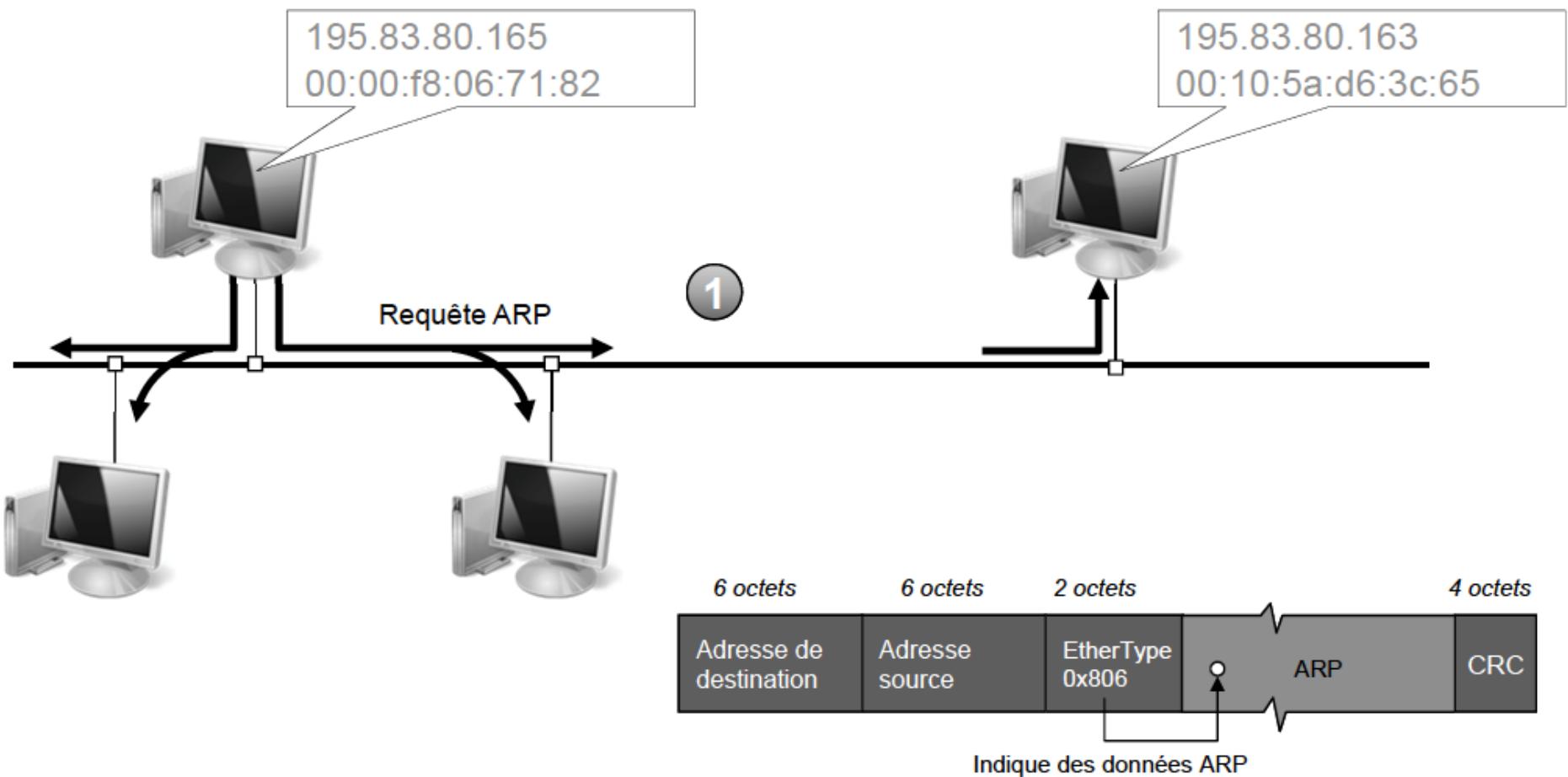


5. ARP (Address Resolution Protocol)

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

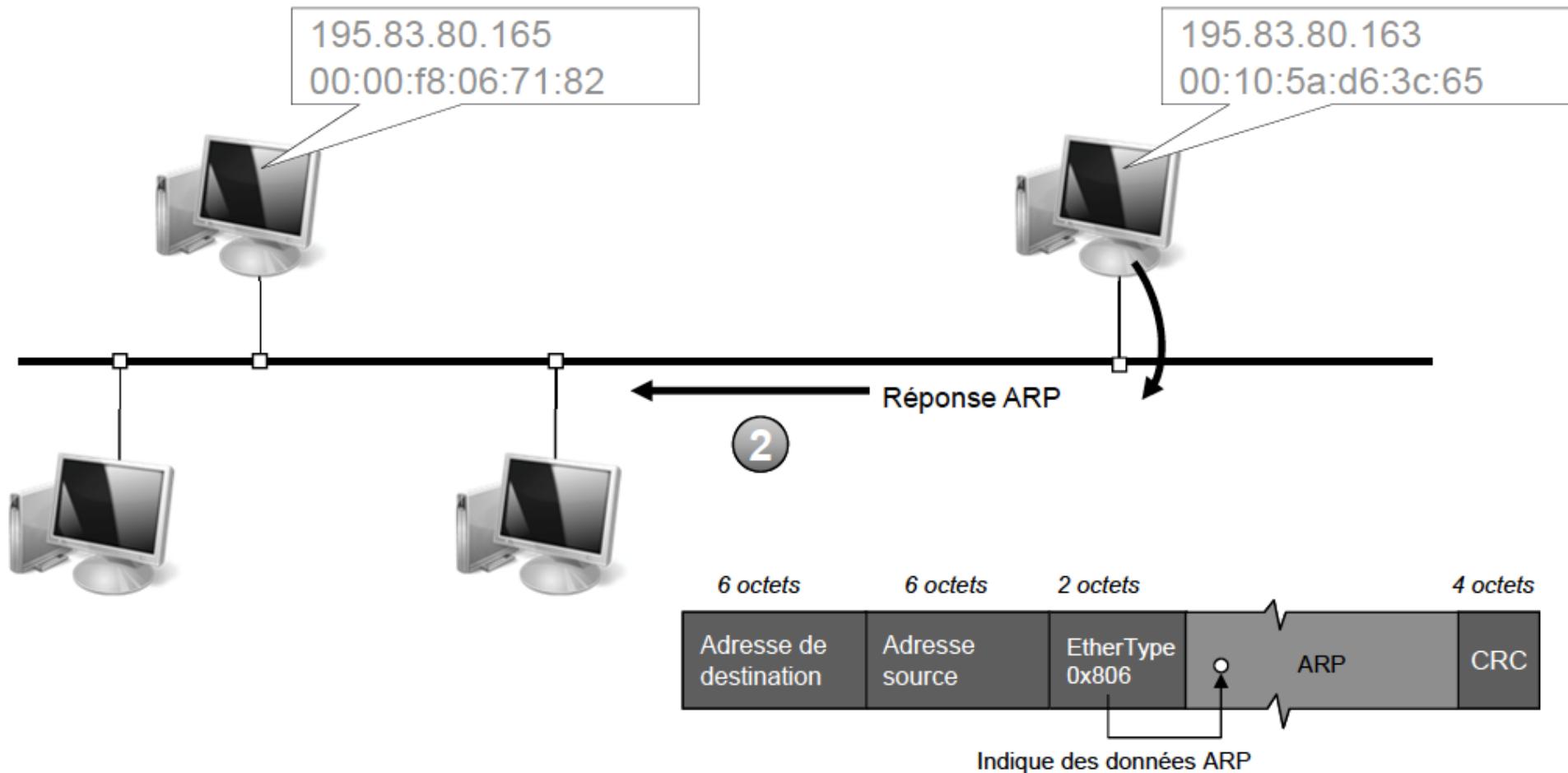
- Au boot d'une machine sans disque
 - on connaît l'adresse Ethernet
 - on ne connaît pas l'adresse IP
- **Protocole RARP (Reverse ARP)**

5. 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
```

5. ARP (Address Resolution Protocol)

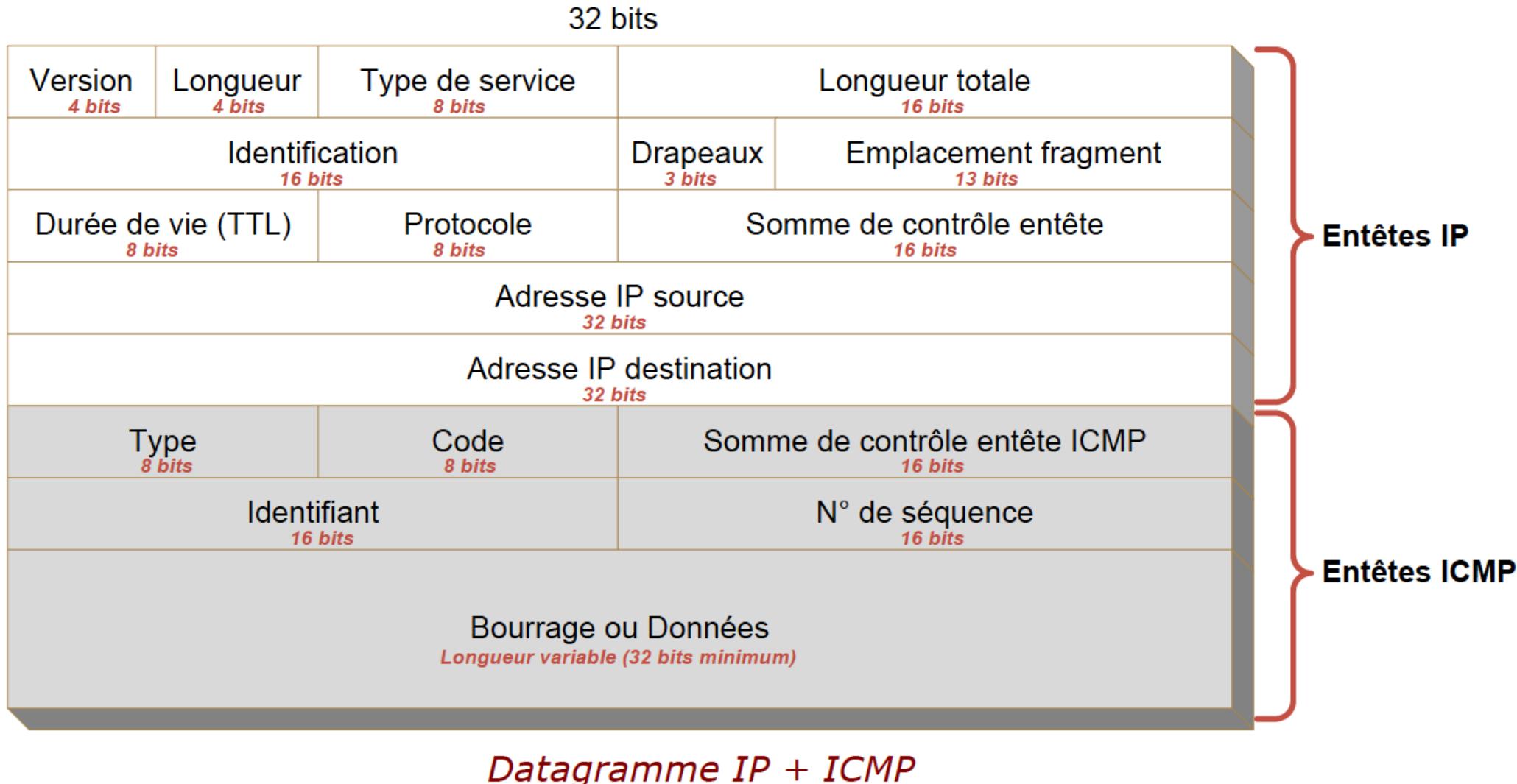


```
1 00:00:f8:06:71:82 ff:ff:ff:ff:ff:ff ARP Who has 195.83.80.163? Tell 195.83.80.165
2 00:10:5a:d6:3c:65 00:00:f8:06:71:82 ARP 195.83.80.163 is at 00:10:5a:d6:3c:65
```

5. 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 on ne connaît pas l'adresse Ethernet
 - 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 on connaît l'adresse Ethernet
- Messages ICMP encapsulés dans des datagrammes IP

5. ICMP (Internet Control Message Protocol)



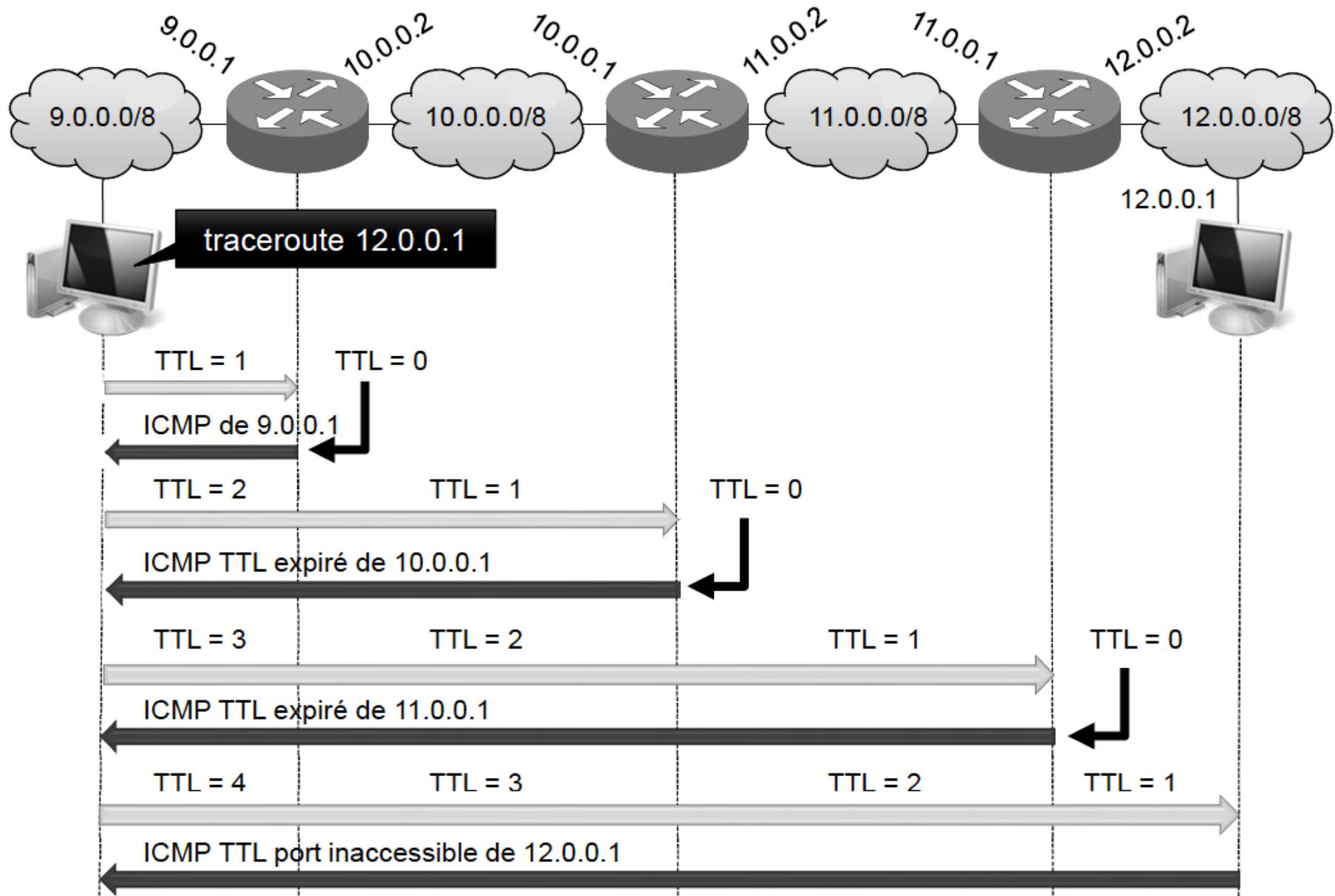
5. ICMP (Internet Control Message Protocol)

- 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
- La commande **traceroute**
 - envoie un paquet UDP avec un TTL (time-to-live) 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

5. ICMP - ping

```
[Minhs-MBP:~ minhtan$ ping google.fr
PING google.fr (216.58.213.131): 56 data bytes
64 bytes from 216.58.213.131: icmp_seq=0 ttl=53 time=20.828 ms
64 bytes from 216.58.213.131: icmp_seq=1 ttl=53 time=20.149 ms
64 bytes from 216.58.213.131: icmp_seq=2 ttl=53 time=22.598 ms
64 bytes from 216.58.213.131: icmp_seq=3 ttl=53 time=20.145 ms
64 bytes from 216.58.213.131: icmp_seq=4 ttl=53 time=20.496 ms
64 bytes from 216.58.213.131: icmp_seq=5 ttl=53 time=20.567 ms
^C
--- google.fr ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 20.145/20.797/22.598/0.840 ms
Minhs-MBP:~ minhtan$
```

5. ICMP - traceroute



5. ICMP - traceroute

```
[Minhs-MBP:~ minhtan$ traceroute google.fr
traceroute to google.fr (216.58.204.99), 64 hops max, 52 byte packets
 1 bbox.lan (192.168.1.254)  5.599 ms  4.954 ms  1.828 ms
 2 * * *
 3 62.34.2.94 (62.34.2.94)  12.518 ms  10.948 ms *
 4 62.34.2.108 (62.34.2.108)  23.949 ms  22.158 ms  20.999 ms
 5 la12.rpt01-ix2.net.bbox.fr (212.194.171.86)  19.661 ms  19.816 ms  20.147 ms
 6 72.14.213.208 (72.14.213.208)  26.435 ms  22.319 ms  21.607 ms
 7 * * *
 8 72.14.237.92 (72.14.237.92)  20.369 ms  19.319 ms
 216.239.48.26 (216.239.48.26)  20.854 ms
 9 108.170.245.5 (108.170.245.5)  20.307 ms
 108.170.235.37 (108.170.235.37)  20.499 ms
 par10s28-in-f99.1e100.net (216.58.204.99)  21.392 ms
[Minhs-MBP:~ minhtan$ traceroute 127.0.0.1
traceroute to 127.0.0.1 (127.0.0.1), 64 hops max, 52 byte packets
 1 localhost (127.0.0.1)  0.644 ms  0.050 ms  0.032 ms
Minhs-MBP:~ minhtan$ ]
```

6. Protocoles TCP et UDP

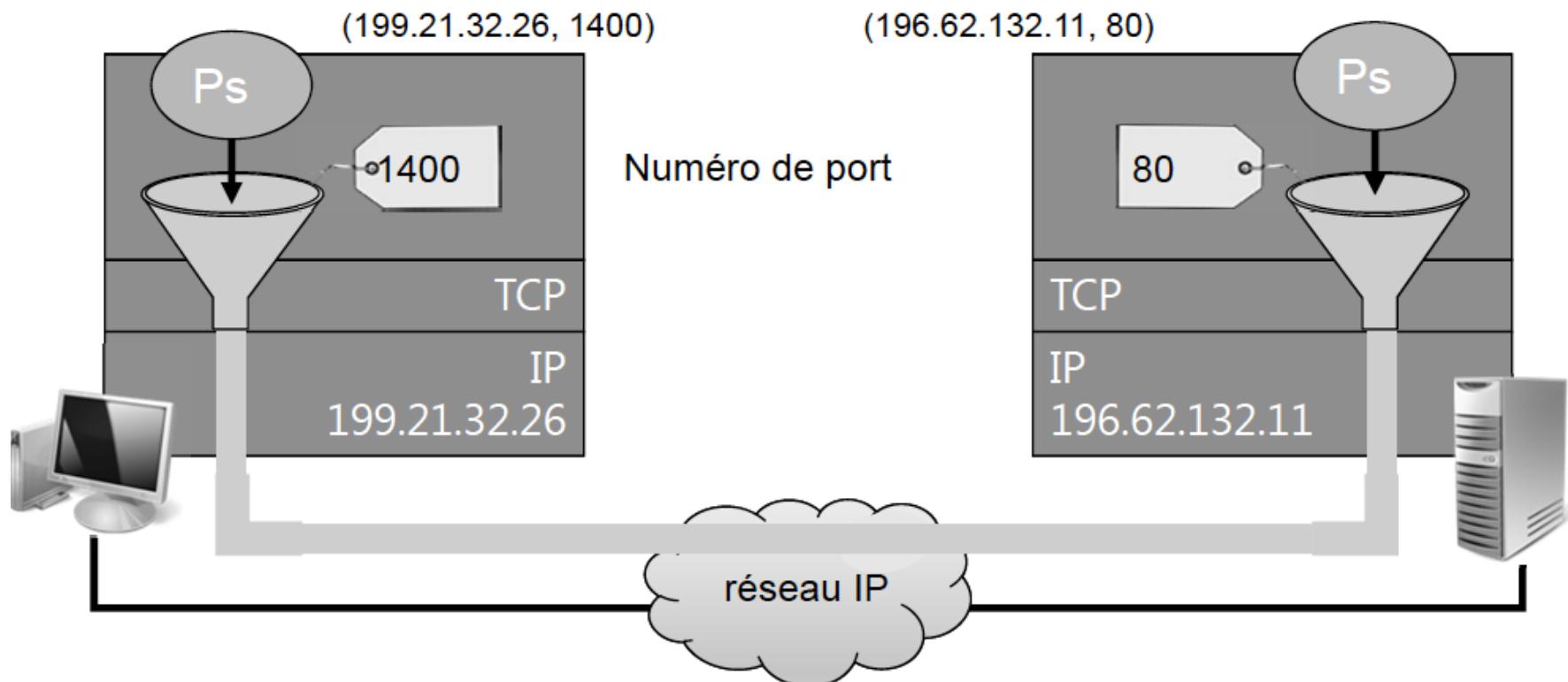
6. Couche Transport

Deux protocoles pour la communication entre applications

- UDP (User Datagram Protocol)
 - mode sans connexion
 - pas de contrôle d'erreur (sans garantie)
- TCP (Transmission Control Protocol)
 - protocole orienté connexion
 - offre de la fiabilité (pas de perte, pas d'erreur)
 - ordonné
 - contrôle de flux

6. Couche Transport

- Identification d'une application par un numéro de port
- **socket** : combinaison d'une adresse IP et d'un numéro de port
- Protocole de transport : combinaison 2 sockets → une connexion TCP/un échange UDP <*protocole, @IP local, n°port local, @IP distance, n°port distance*>



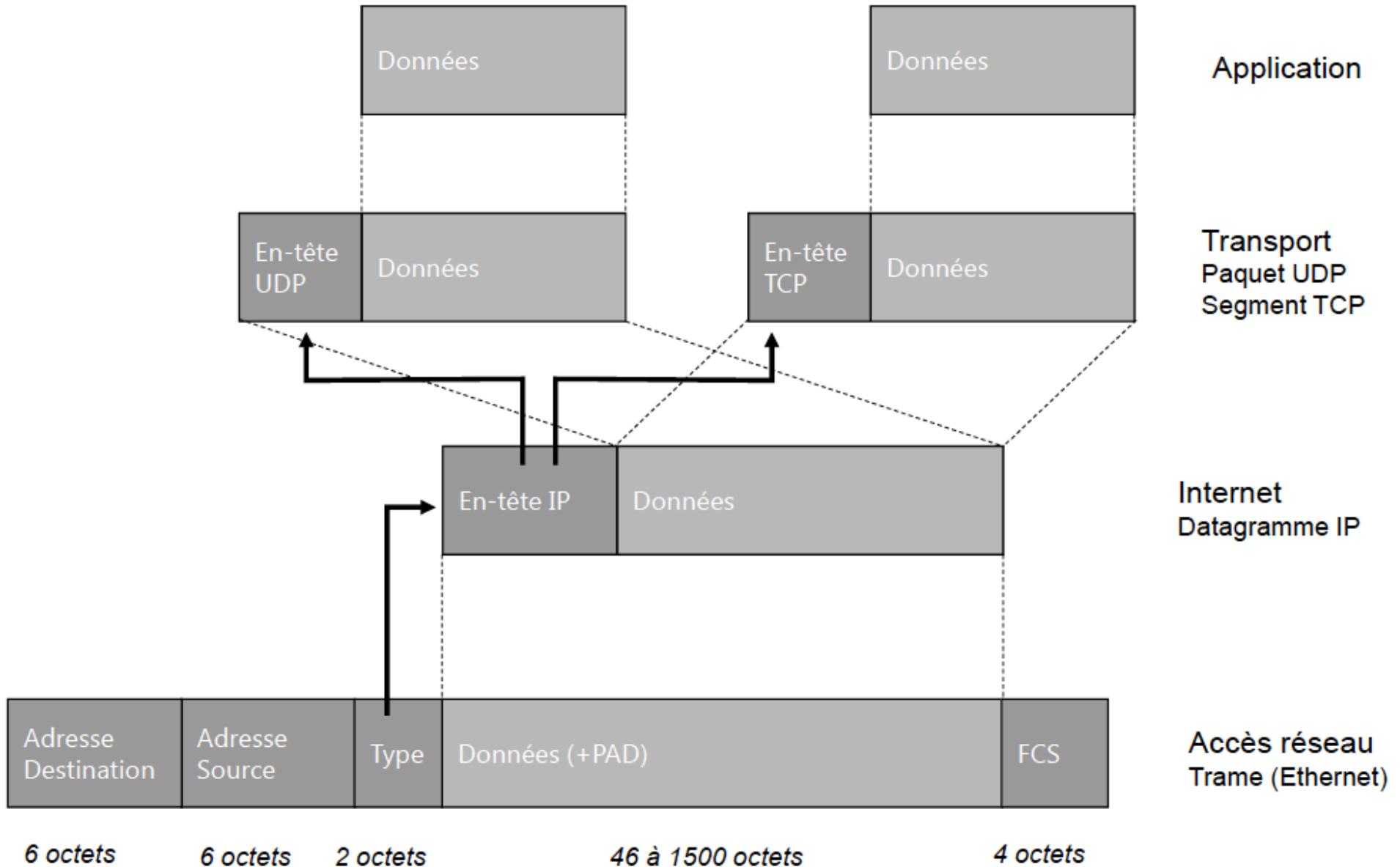
6. Notion de port

- 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
- 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**

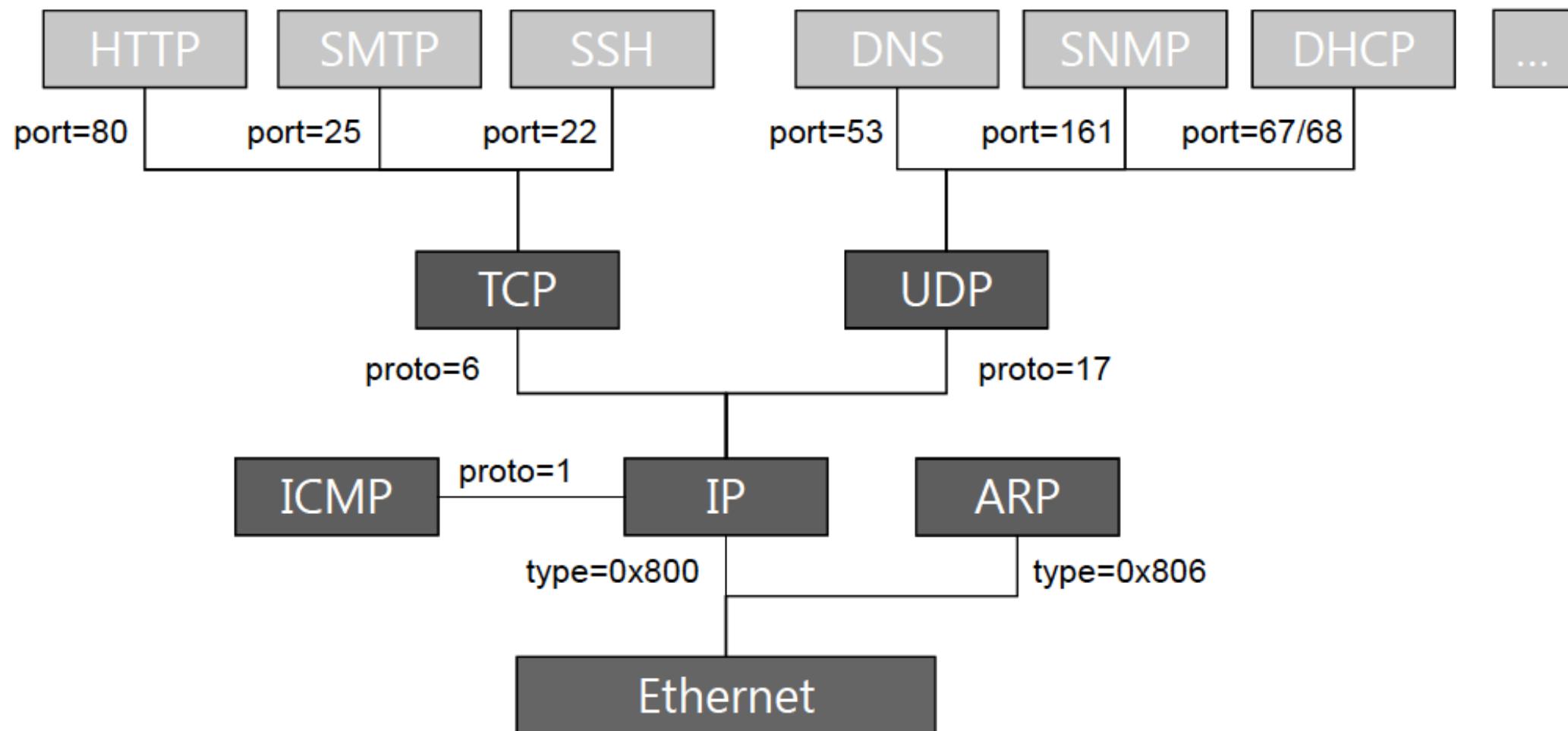
6. Notion de port

Port	Service	Port	Service
20 et 21	FTP	989 et 990	FTPS
22	SSH		
23	Telnet		
25	SMTP		
53	DNS		
80	Web	443	HTTPS
110	POP3	995	POP3S
143	IMAP	993	IMAPS
161	SNMP		
179	BGP		
520	RIP		

6. Encapsulation : rappel



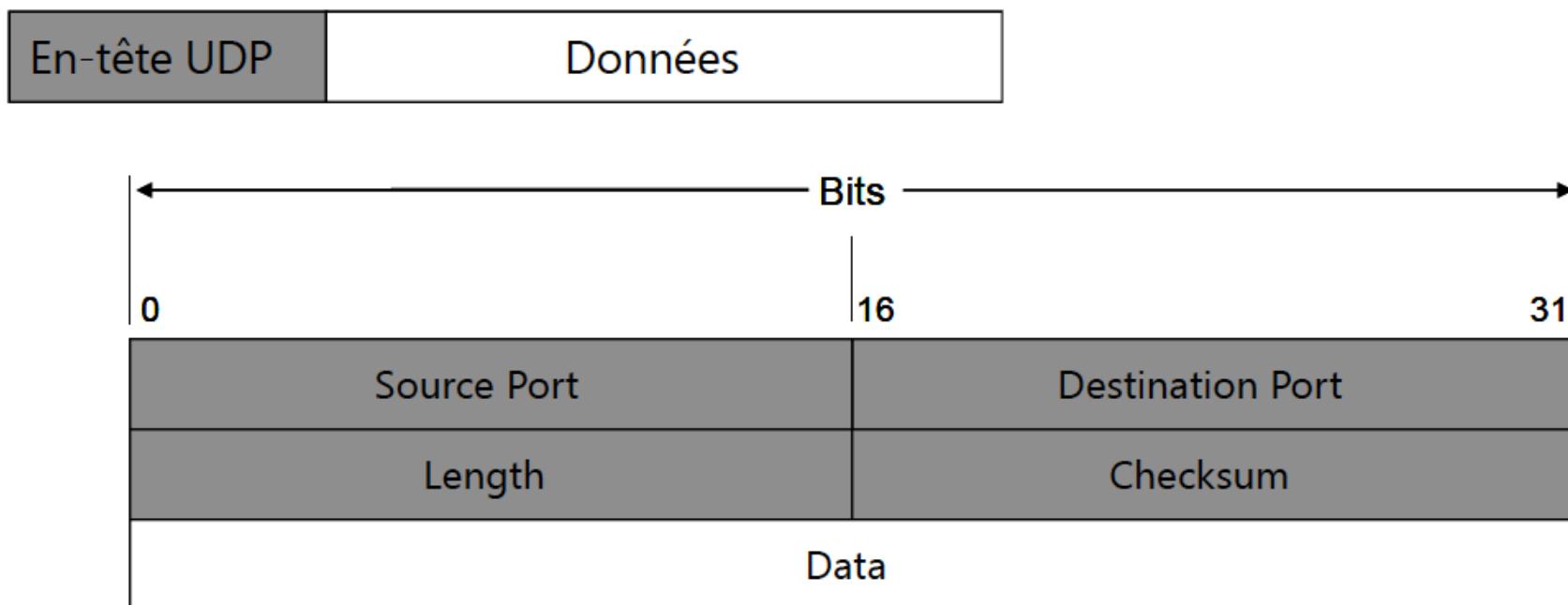
6. Identification des protocoles



6. UDP

- **UDP (User Datagram Protocol) :**

- service sans connexion, sans garantie (non fiable) et beaucoup plus simple que TCP
- aucune valeur ajoutée par rapport aux services de IP
- utilisé pour transmettre une faible quantité de données sans accusé de réception



6. UDP

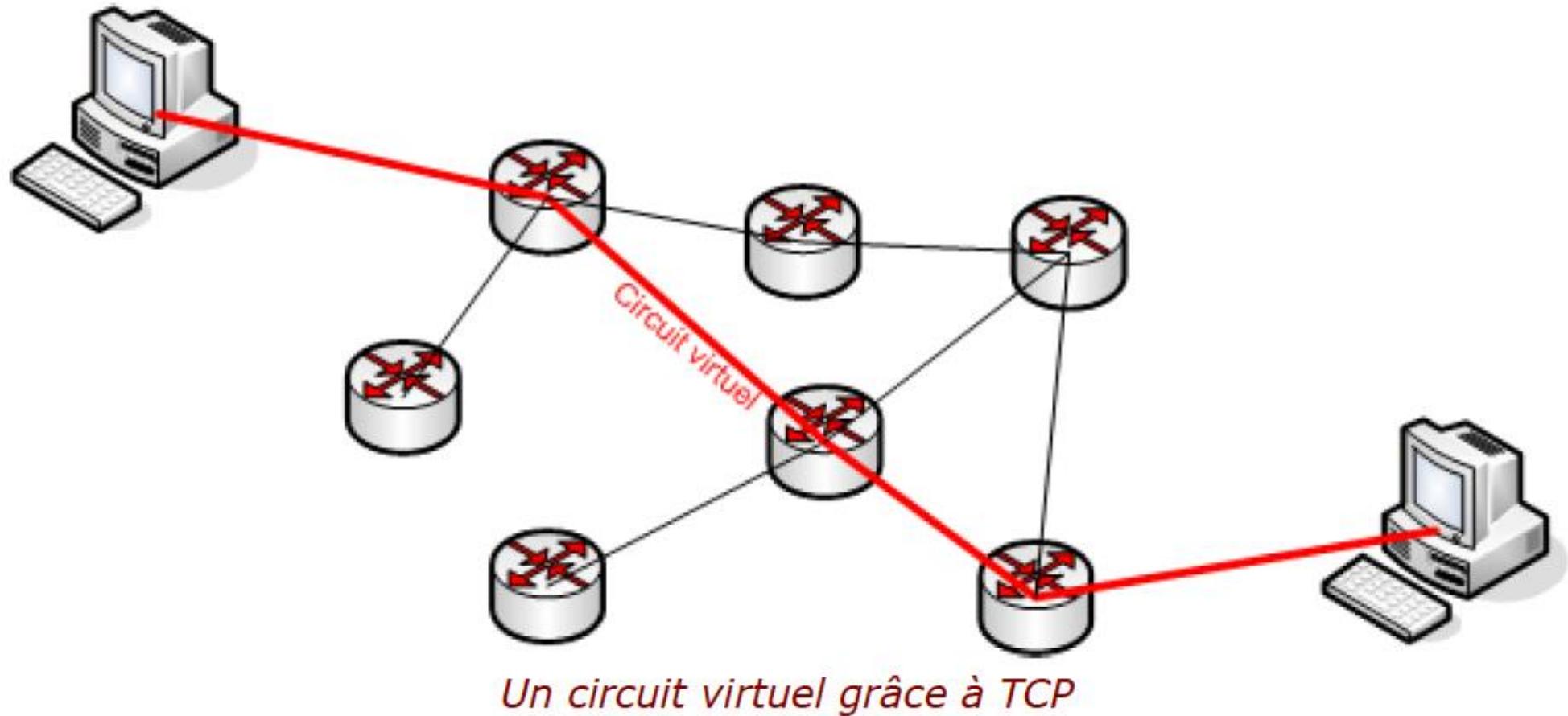
- Champ ***Length***
 - contient la longueur du paquet UP en octets (en-tête + données)
 - une valeur minimale de 8 (données vide)
- Champ ***Checksum***
 - contrôle d'erreur (16 bits) → validité du paquet de la couche Transport
 - calculé sur un pseudo en-tête constitué
 - des adresses IP source et destinations, du code de protocole (17) et de la longueur du paquet UDP
 - en-tête UDP elle même
 - des données

6. TCP

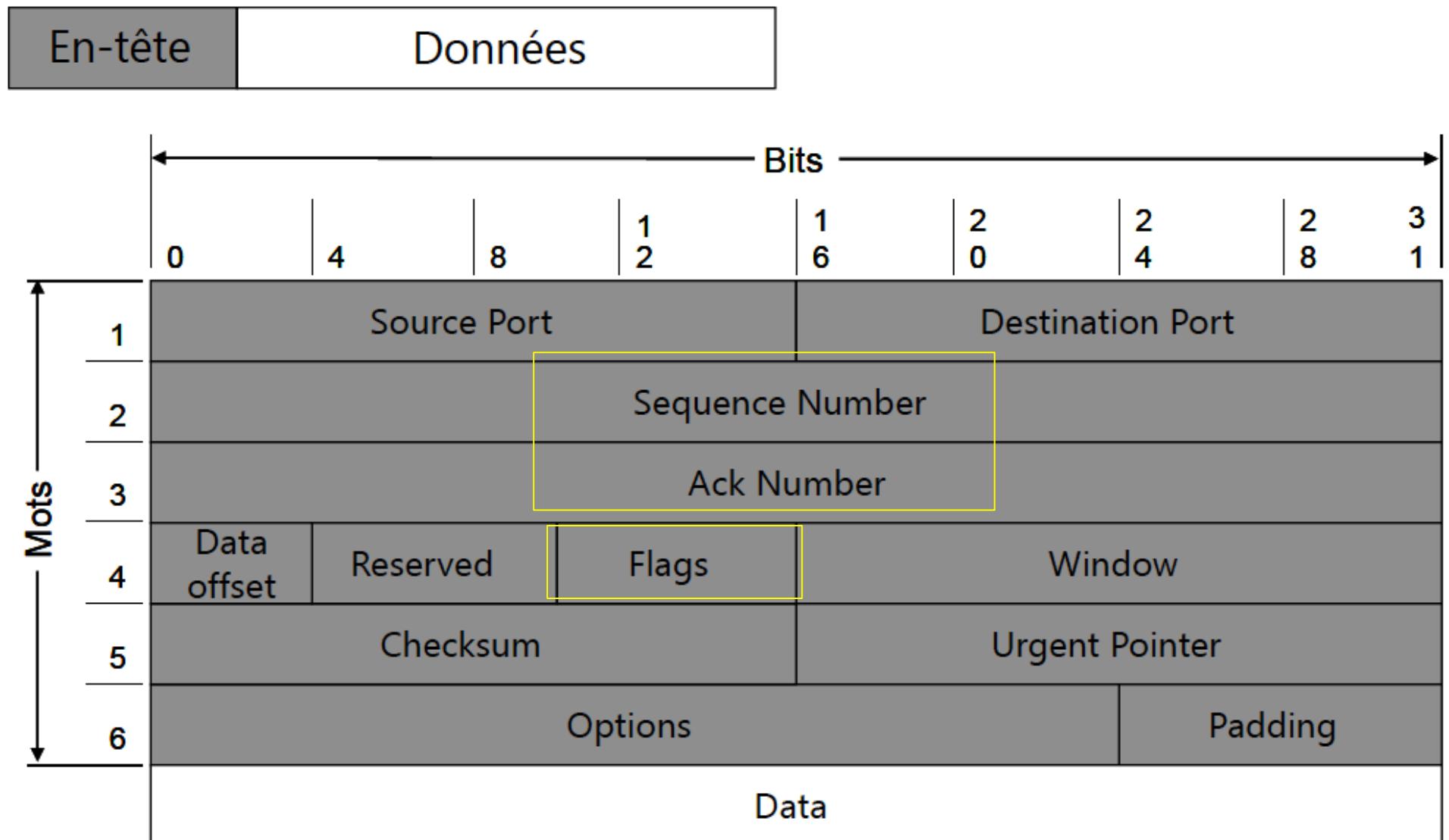
- **TCP (Transport Control Protocol)** : n'est exécuté que par la machine source et la machine destination (pas dans les routeurs)
- Caractéristiques :
 - transport de bout en bout
 - mode connecté : ouverture, fermeture, gestion de connexion
 - sans erreur : contrôle et retransmission si nécessaire
 - sans perte : “numérotation” et retransmission
 - ordonnée : préservation du séquencement
 - système d'acquittement
 - contrôle de flux
 - identification du service par numéro de port

6. TCP

- Rappel : mode connecté avec circuit virtuel



6. Format du segment TCP

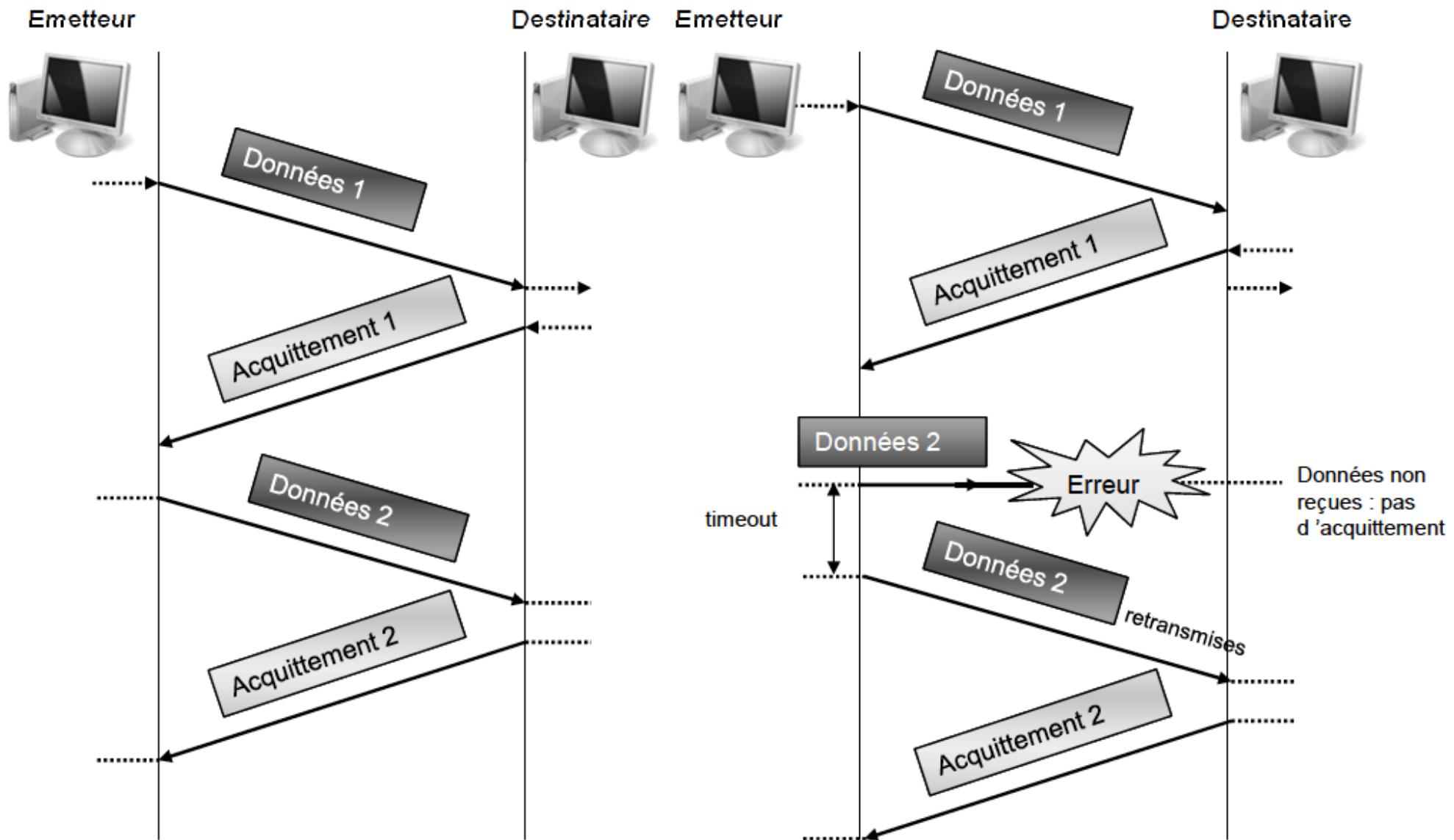


6. Fiabilité de transmission TCP

Fiabilité TCP → assurée par un mécanisme appelé **PAR** (Positive Acknowledgment with Retransmission)

- Un numéro de séquence (**Sequence Number**) est attribué à chaque paquet 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

6. Mécanisme PAR



6. 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 à 3 temps** (three-way handshake), est échangée entre les deux extrémités
- Ce mécanisme met en œuvre 3 échanges de segment 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)

6. Mécanisme de la poignée de main

Pour l'établissement de la connexion :

Station A



Station B



1^{ère} étape

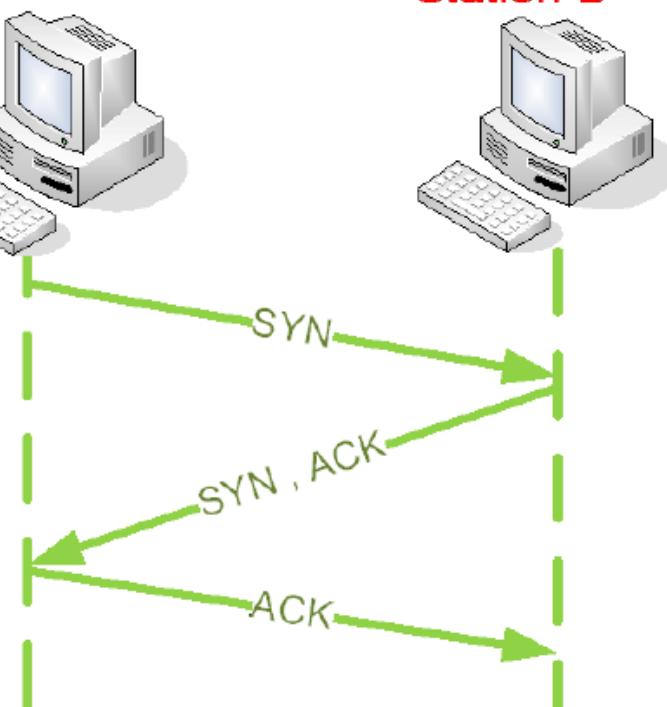
La station 1 émet un segment avec le drapeau SYN (Demande de connexion).

2^{ème} étape

La station 2 répond en émettant un segment avec le drapeau SYN/ACK (SYN : La station B envoie elle aussi une demande d'ouverture de connexion. ACK : Accusé de réception de la demande de connexion précédente).

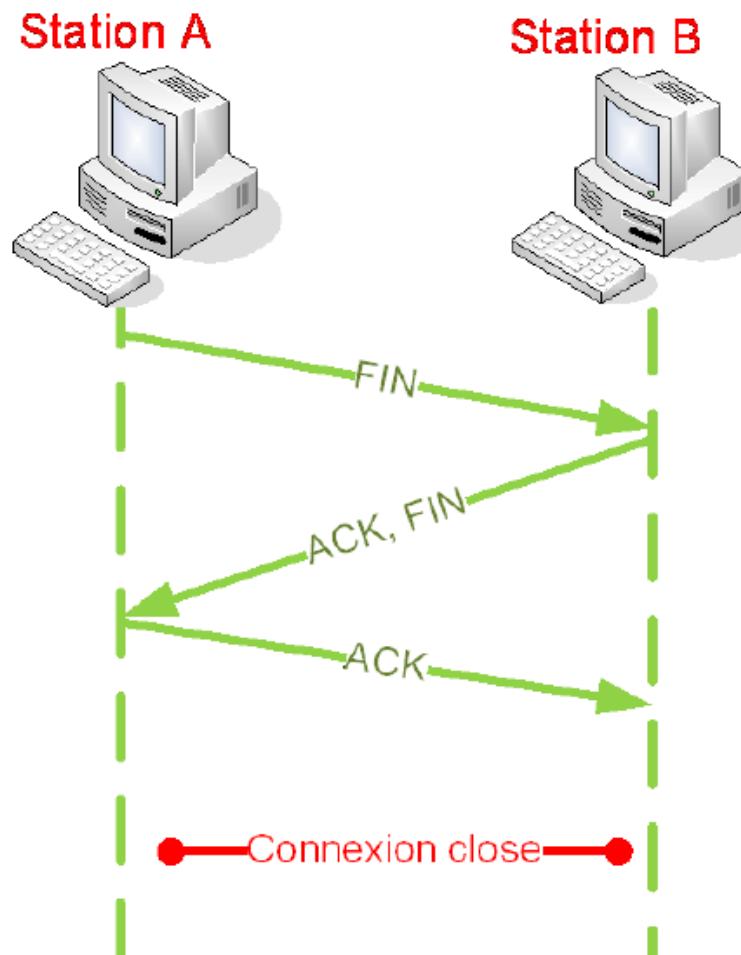
3^{ème} étape

La station 1 répond par un segment avec le drapeau ACK précisant qu'elle a reçu la confirmation de la demande de connexion



6. Mécanisme de la poignée de main

Pour la fermeture de la connexion, le flag **FIN** est utilisé :



1^{ère} étape

La station qui souhaite mettre fin à la connexion envoie le drapeau FIN.

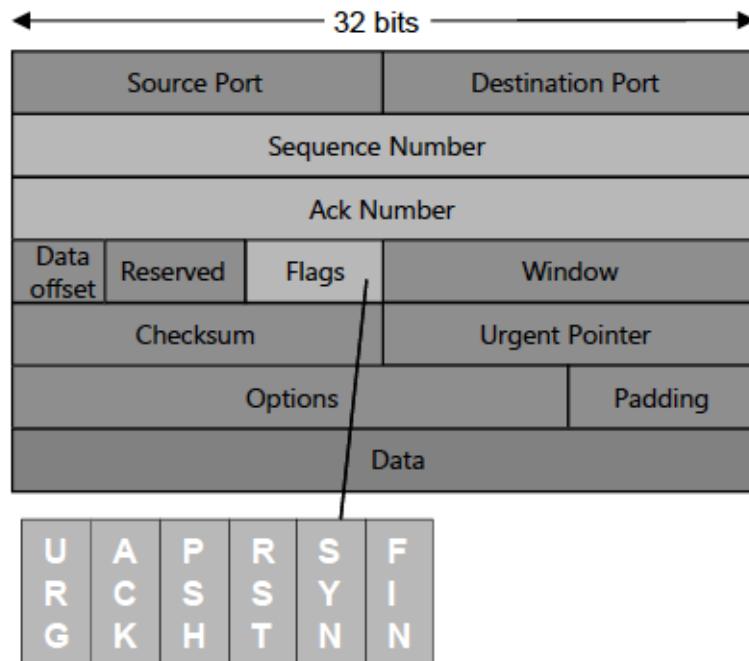
2^{ème} étape

La station qui reçoit le drapeau FIN acquitte avec le drapeau ACK puis envoie aussi le drapeau FIN. Selon les cas, l'envoi de ACK et FIN se fait dans 2 segments différents

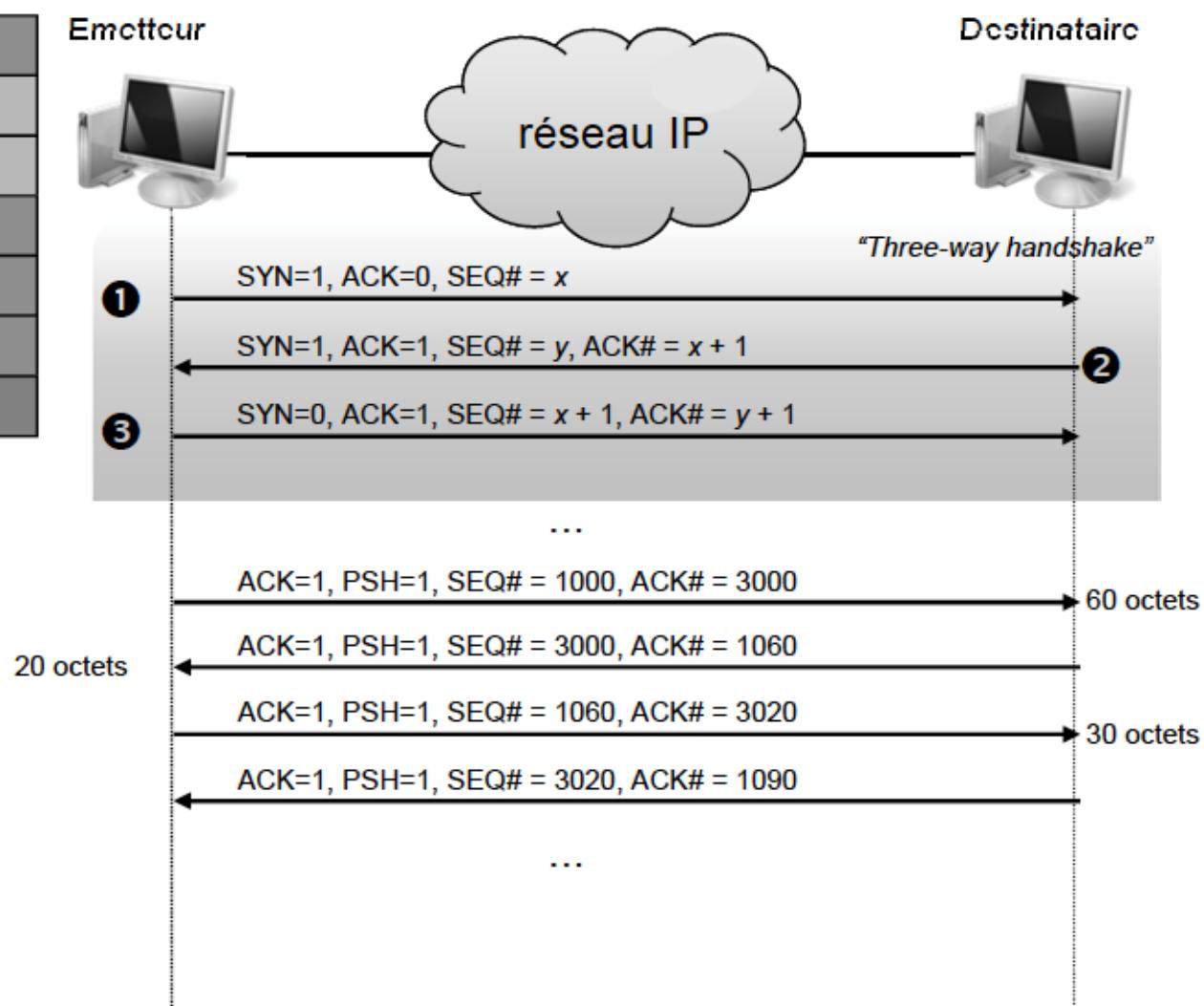
3^{ème} étape

La station A acquitte la demande de FIN

6. Mécanisme de la poignée de main

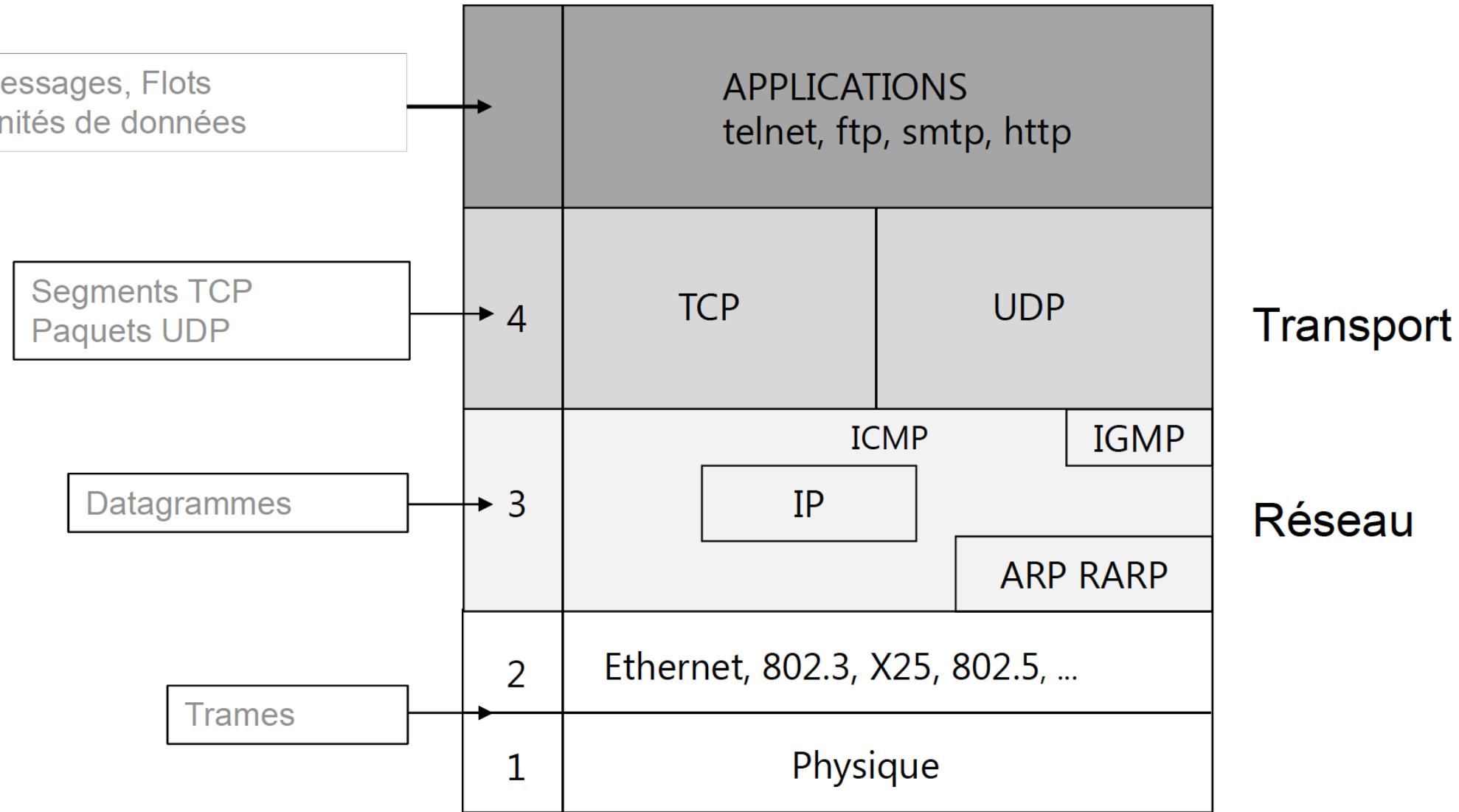


- **URG** : tenir compte du champ *Urgent Pointer*
- **ACK** : tenir compte du champ *Acknowledgment Number*
- **PSH** : délivrer immédiatement les données au processus de couche supérieure
- **RST** (reset) : réinitialiser la connexion
- **SYN** : établir une connexion
- **FIN** : terminer la connexion



7. Services réseau

7. Services réseau (applications)



7. Services réseau (applications)

❖ La communication

- Courrier électronique :
 - **SMTP** (Simple Mail Transfer Protocol) – port 25
 - **IMAP** (Internet Message Access Protocol) – port 143
 - **POP3** (Post Office Protocol) – port 110
- Les nouvelles : **NNTP** (Network News Transfer Protocol) – port 119
- **Efnet** : canaux de discussion IRC (Internet Relay Chat)
- **Talk**
- **VoIP** (Voice over IP) et **ToIP** (Telephony over IP)

7. Services réseau (applications)

❖ Le transfert de fichier

- **FTP (File Transfer Protocol)**
 - Utilise 2 connexions TCP → transfert fiable
 - 1 de contrôle (commandes et réponses) : port 21
 - 1 de transfert de données : port 20
 - Mode client/serveur

7. Services réseau (applications)

❖ La prise de commande à distance

- Protocole **TELNET** (Terminal Network protocol)
 - Terminal virtuel, remote terminal, terminal à distance
 - Utilise une connexion TCP
 - Fiable mais gourmand en bande passante
 - Utilise le port 23 pour le serveur
- Protocole **SSH** (Secure Shell)
 - Utilise une connexion TCP (port 22 pour le serveur)

7. Services réseau (applications)

❖ Le Web

- Protocole **HTTP** (Hypertext Transfer Protocol) – port 80
 - chaque demande d'accès à un URL (Uniform Ressource Locator), une requête HTTP est émise
 - HTTP gère la totalité des échanges réalisées entre un client et un serveur
- Le **DNS** (Domaine Name System) – port 53
 - Correspondance entre un nom et une adresse IP
 - Noms plus faciles à retenir que les adresses IP
 - Hiérarchie de serveurs

Référence

- <https://fr.wikipedia.org/>
- <https://fr.wikibooks.org/>
- A. Tanenbaum, *Computer network* (2nd ed), Prentice-hall Int. Ed.
- L. Petrucci, *Principes et architecture des réseaux*, <https://lipn.univ-paris13.fr/~petrucci/>
- P. Felix, *Architecture des réseaux informatiques*, <http://dept-info.labri.fr/~felix/>
- B. Petit et al., *Architecture des réseaux: cours et exercices corrigés*, Ellipses, 2017
- S. Cateloin et al., *Mini Manuel des réseaux informatique*, Dunod, 2012
- D. Mercier, *Routage et Routage dans l'Internet*, IUT R&T, 2008-2009
- C. Bulfone, *La pile TCP/IP*, Master MIASHS/DCISS
- M.A. Peraldi-Frati, *Cours L3 Informatique Réseaux*, LPSIL ADMIN
- D. Dromard, D. Seret, *Architecture des réseaux*, PERSON EDUCATION, 2009