Introduction à la cyberstructure de l'internet - réseaux et sécurité

UE CNAM - UTC505

François Lacomme <francois.lacomme@2isa.net>

Documents liés aux cours et TP : https://utc505.seancetenante.com

2024/2025

1 - Plan du cours

- Diviser pour régner **(modèle OSI)**
  - Découverte de l'**architecture de communication en couches**. Du modèle OSI à l'architecture Internet; introduction aux protocoles http, DNS et à l'outil d'analyse de traces Wireshark.
- Les autoroutes de l'information : nids de poules et travaux en tous genres **(couche physique)**
  - Concepts et problèmes de la transmission de données : erreurs de transmission, le contrôle d'erreur, notion de bande passante, traitement des signaux, atténuation, modulation, multiplexage, commutation, synchronisation d'horloge, problèmes de caractère et de bit stuffing.
- Collectivisme ou Libre entreprise… à la recherche d'un modèle équitable **(sous-couche MAC)**
  - Grandes familles de protocoles à compétition et à coopération, détail sur CSMA/CD et CSMA/CA en mode infrastructure. Ponts et commutation.

1 - Plan du cours

- Croisements et Destination **(couche réseau)**
  - Adressage, tables de routage et l'expédition de données dans le réseau IP. Evolution de IPv4 à IPv6.
- Une lettre ou un appel ? **(couche transport)**
  - Transport de données entre un client et un serveur à travers UDP et TCP avec le modèle datagramme, et les approches connecté et non connecté. Gestion et utilisation de l'API socket.
- Où sont les clefs ? **(Introduction à la sécurité)**
  - Aspects sécurité de base pour la confidentialité, l'intégrité, l'authentification et la notarisation : principes de cryptographie symétrique et asymétrique, fonctions de hachage cryptographique.

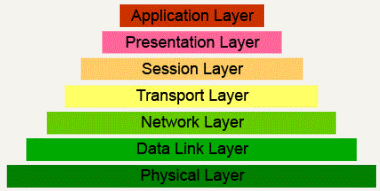
2 - Prologue

Taille de réseaux

Taille		Abréviation	Réseau
1m		PAN Personal Area Network	Réseau domestique
10 m	salle		
100 m	entreprise	LAN Local Area Network	Réseau local
1 km		MAN Metropolitan Area Network	Réseau métropolitain
10 km	ville		
100 km	région	WAN Wide Area Network	Réseau longue distance Réseau étendu
1000 km	pays, continent		
10 000 km	terre entière	GAN (Global Area Network) **Internet**	Réseau mondial ex. : Internet (Réseau de réseaux TCP/IP)

Fig 0.1 - Taille de réseaux

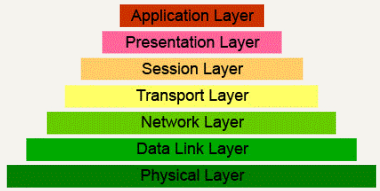
Architecture de communication en couches



1 - Architecture logicielle des réseaux

- **Forte structuration des logiciels de réseaux pour réduire la complexité de conception** des réseaux
  - Diviser pour mieux régner
  - Rendre chaque problème de communication autonome avec des interfaces claires
  - Une approche récursive et générique
- **Organisation en couches ou niveaux :**
  - Une couche N offre un ensemble de **services** à la couche immédiatement au-dessus N+1.
  - La couche N se sert pour cela de la couche N-1 en dessous, masquant à la couche N+1 le fonctionnement et la complexité de la couche N-1.
  - Chaque couche est construite au-dessus de la précédente et gère la communication avec **la couche de même niveau** d'une autre machine
  - Cette communication utilise des règles et des conventions appelées **protocoles**
  - Le **support de transmission** (la couche 0), lié à la couche la plus basse, véhicule réellement la communication

Architecture de communication en couches



1 - Architecture logicielle des réseaux

- **Illustration**
  - L'architecture philosophe/ traducteur/ secrétaire

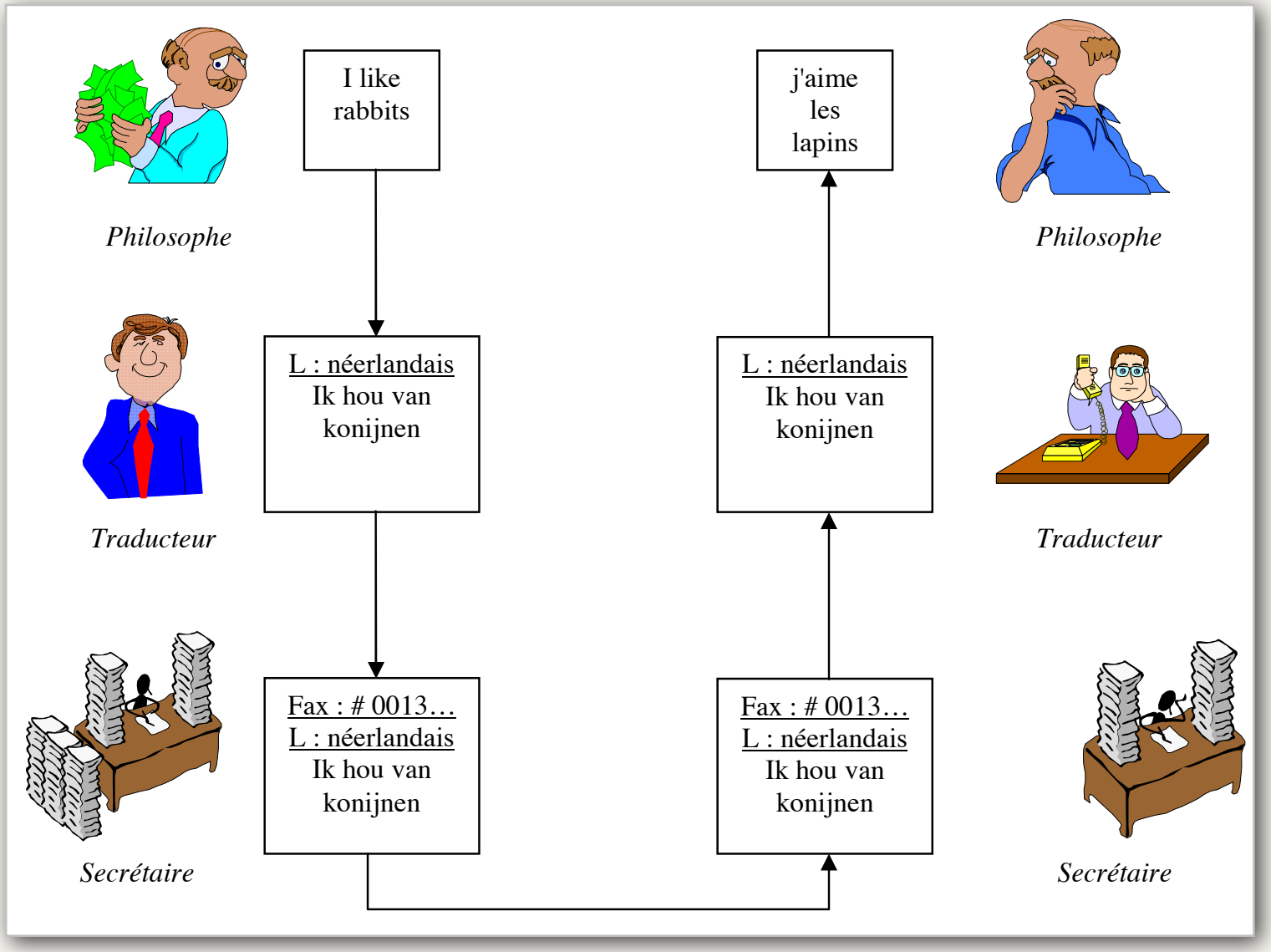
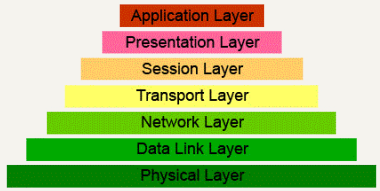


Fig 1.1 - Une architecture à trois couches

Architecture de communication en couches



2 - Définitions

- **Protocole P(n) d'une couche *n***
  - Ensemble des **règles et conventions** utilisées pour le dialogue de la couche **n**
- **Pile de protocoles**
  - Ensemble des protocoles utilisés par un système, avec 1 protocole par couche

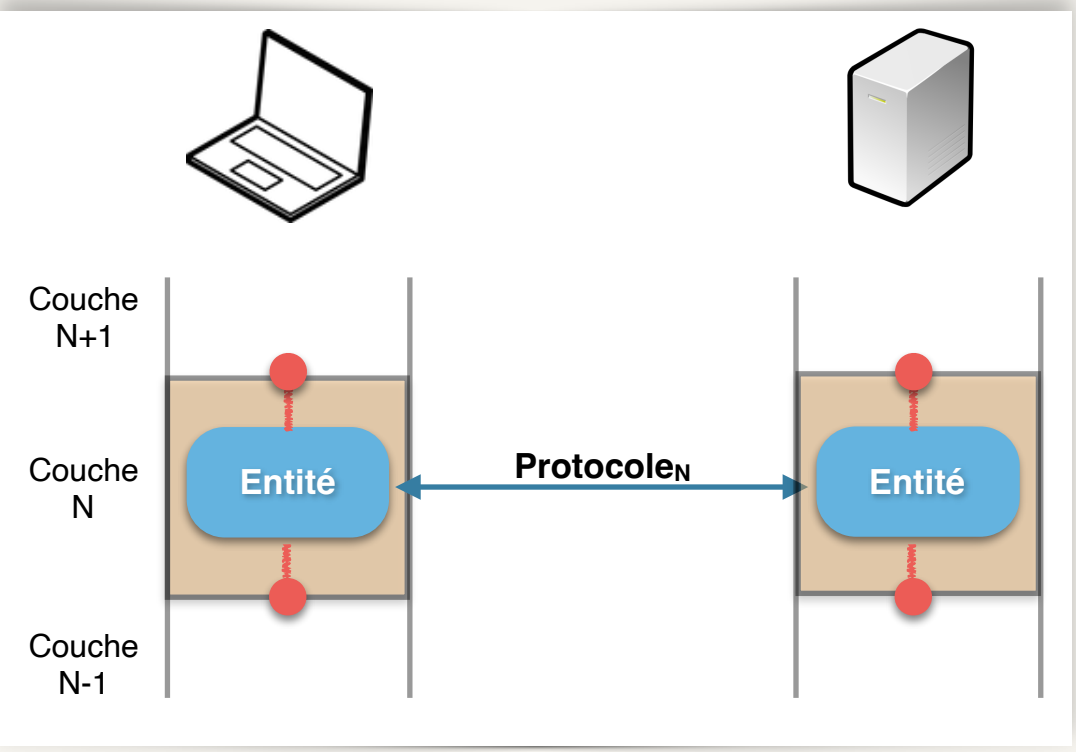


Fig 1.2 - Protocole$_N$ d'une couche N

- **Service**
  - Description abstraite de fonctionnalités à l'aide de **primitives de service** (commandes ou événements)
  - Le service d'une couche ***n*** définit l'ensemble des fonctionnalités possédées par la couche ***n*** et fournies aux entités de la couche ***n+1*** à l'interface ***n/n+1***

2 - Définitions

- **Interface**
  - Accès à l'ensemble des opérations élémentaires et des services qu'une couche (n) offre à la couche (n+1) supérieure
  - Une interface est le moyen concret d'utiliser un **service**

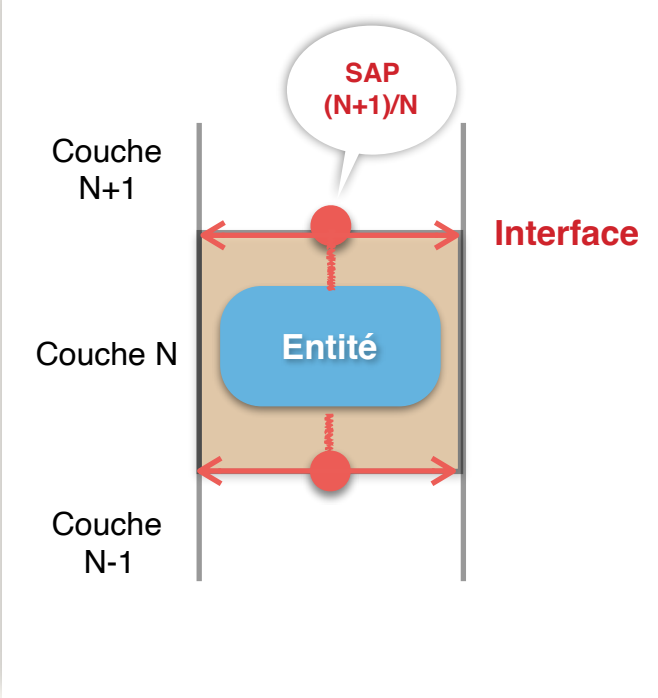
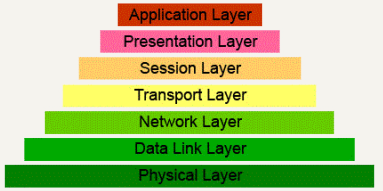


Fig 1.3 - Interfaces entre couches

- **Encapsulation**
  - Technique consistant à ajouter à un bloc de données un en-tête (*header*), et éventuellement une queue (*trailer*). L'en-tête contient les informations de contrôle du protocole.

- **Décapsulation**
  - Extraction des données utiles à partir de l'unité de données de protocole

Architecture de communication en couches



2 - Définitions

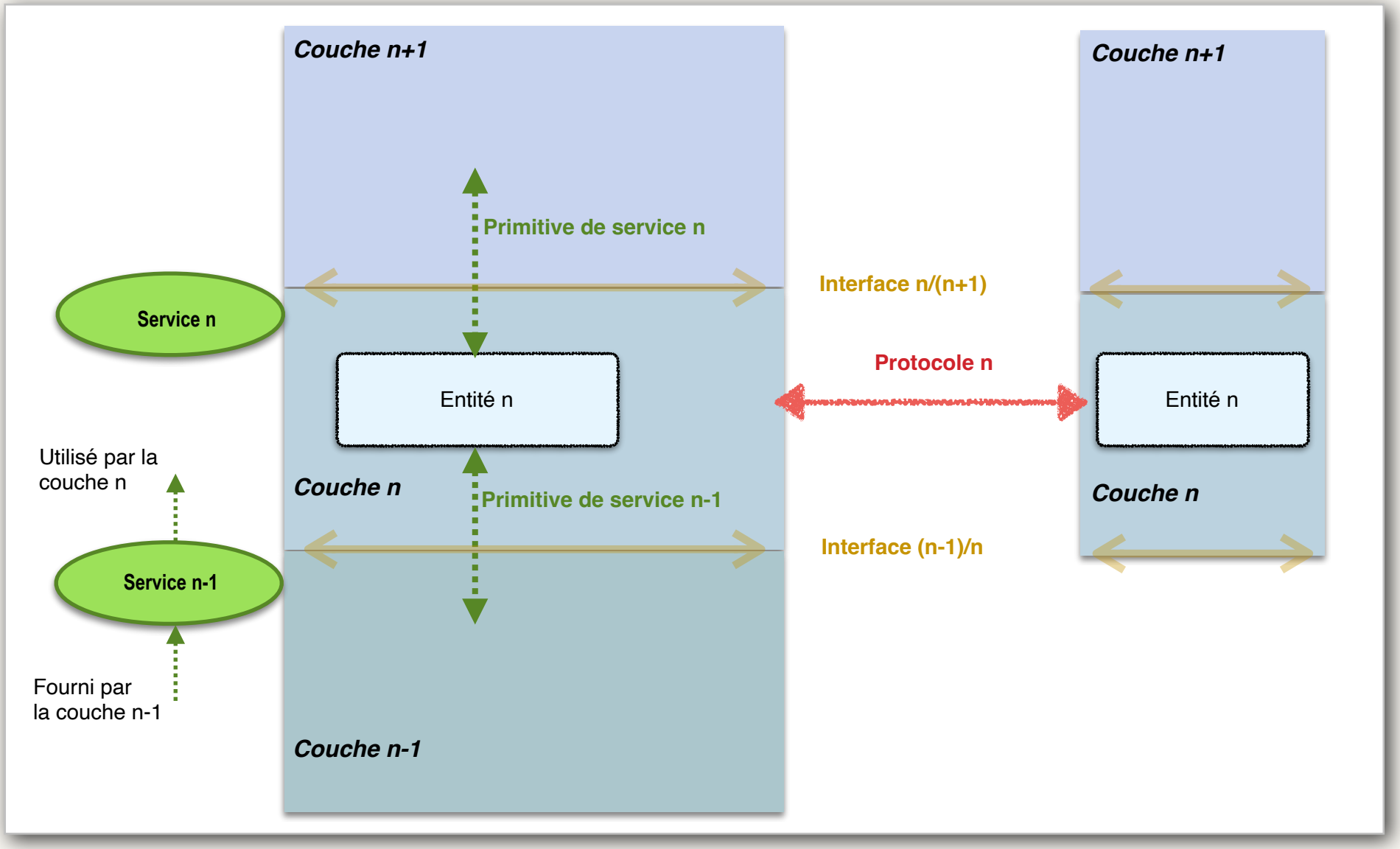
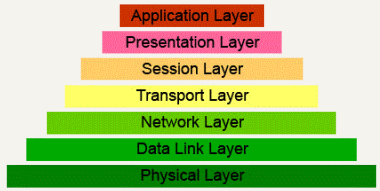


Fig 1.4 - Couches, services et protocoles



2 - Définitions

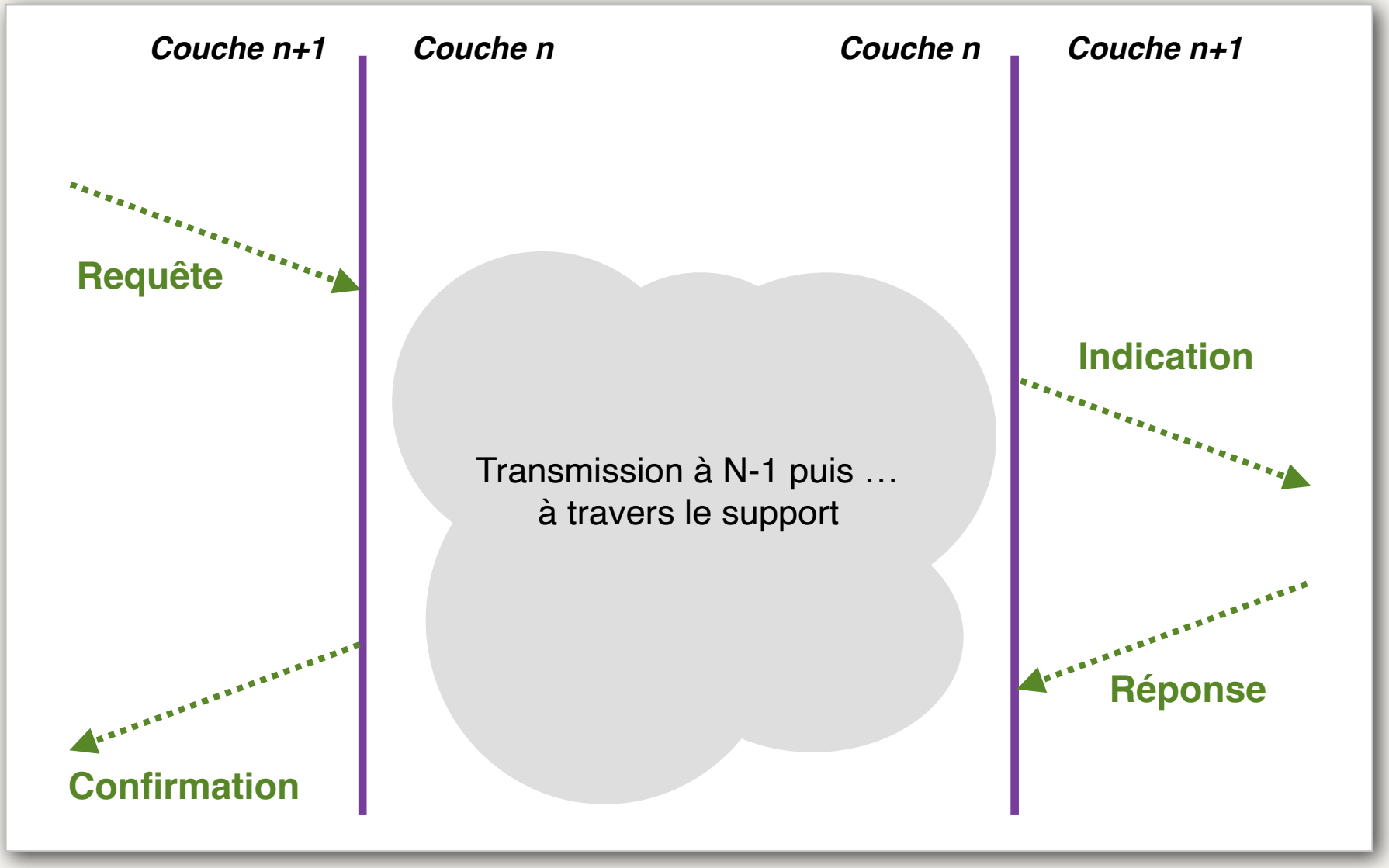
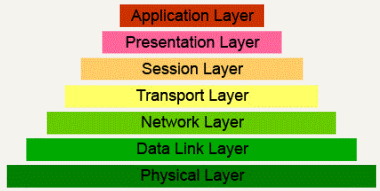


Fig 1.5 - Primitives de service



2 - Définitions

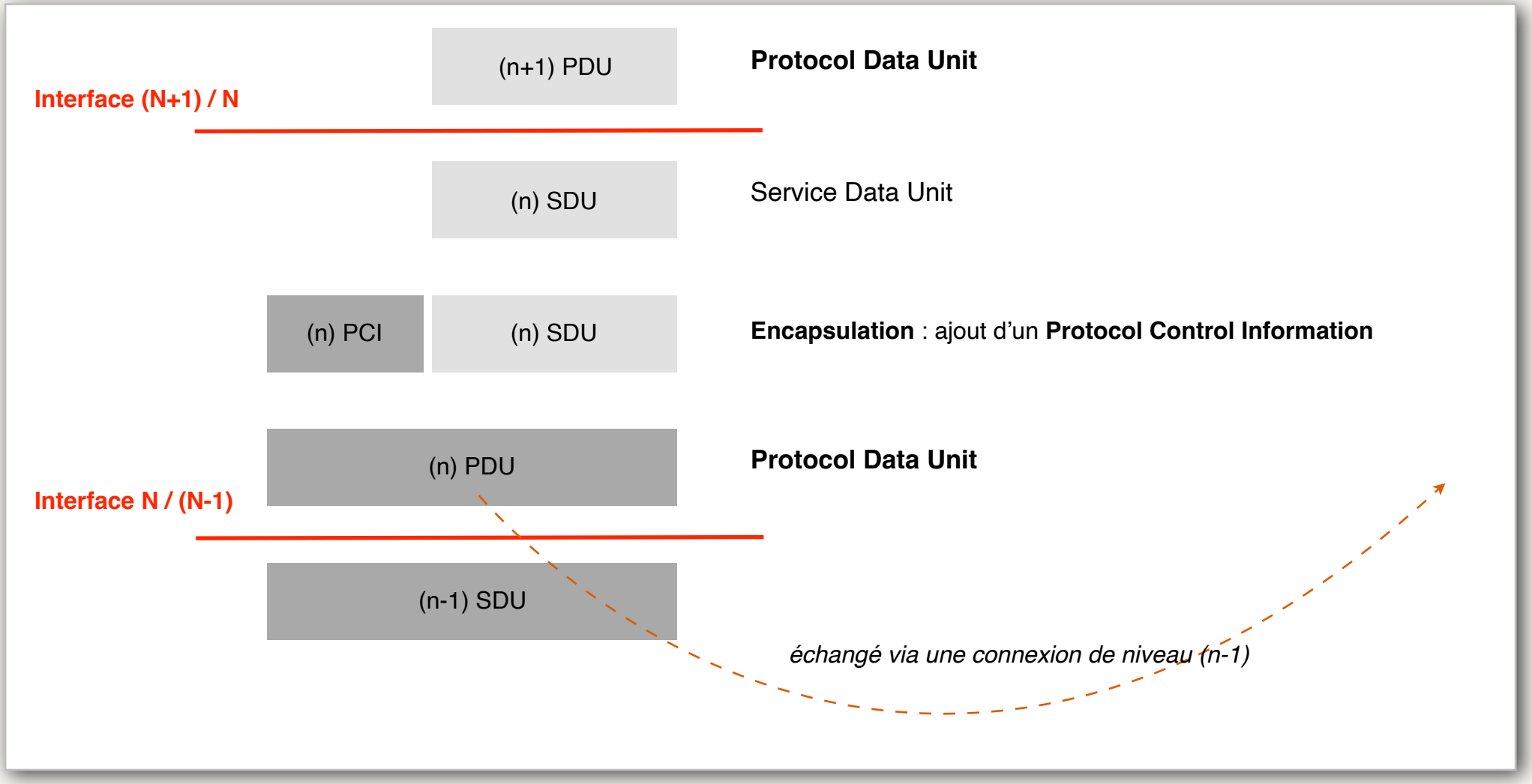


Fig 1.6 - Unités de Données de Protocole & encapsulation

2 - Définitions

- **Modèle OSI**
  - Modèle **OSI**, *Open Systems Interconnection* de l'ISO, *International Standard Organization*

  - Ce modèle de référence a été conçu par l'ISO (années 1970) et a été normalisé en 1984 et révisé en 1994.

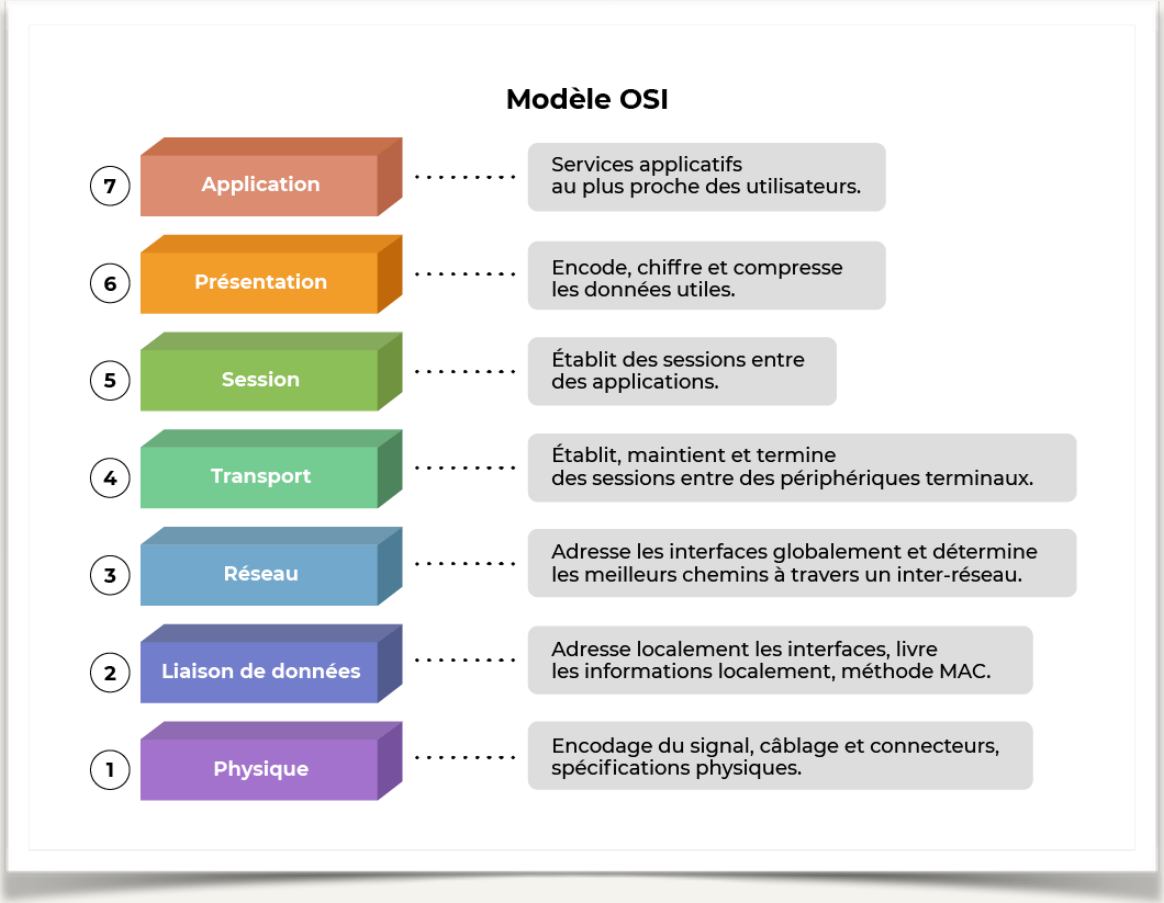
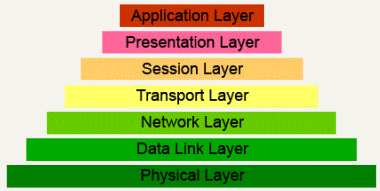


Fig 1.7 - Le modèle OSI

Architecture de communication en couches



3 - Le modèle de référence OSI

- ***P**our*
  ***L**e*
  ***R**éseau*
  ***T**out*
  ***S**e*
  ***P**asse*
  ***A**utomatiquement*

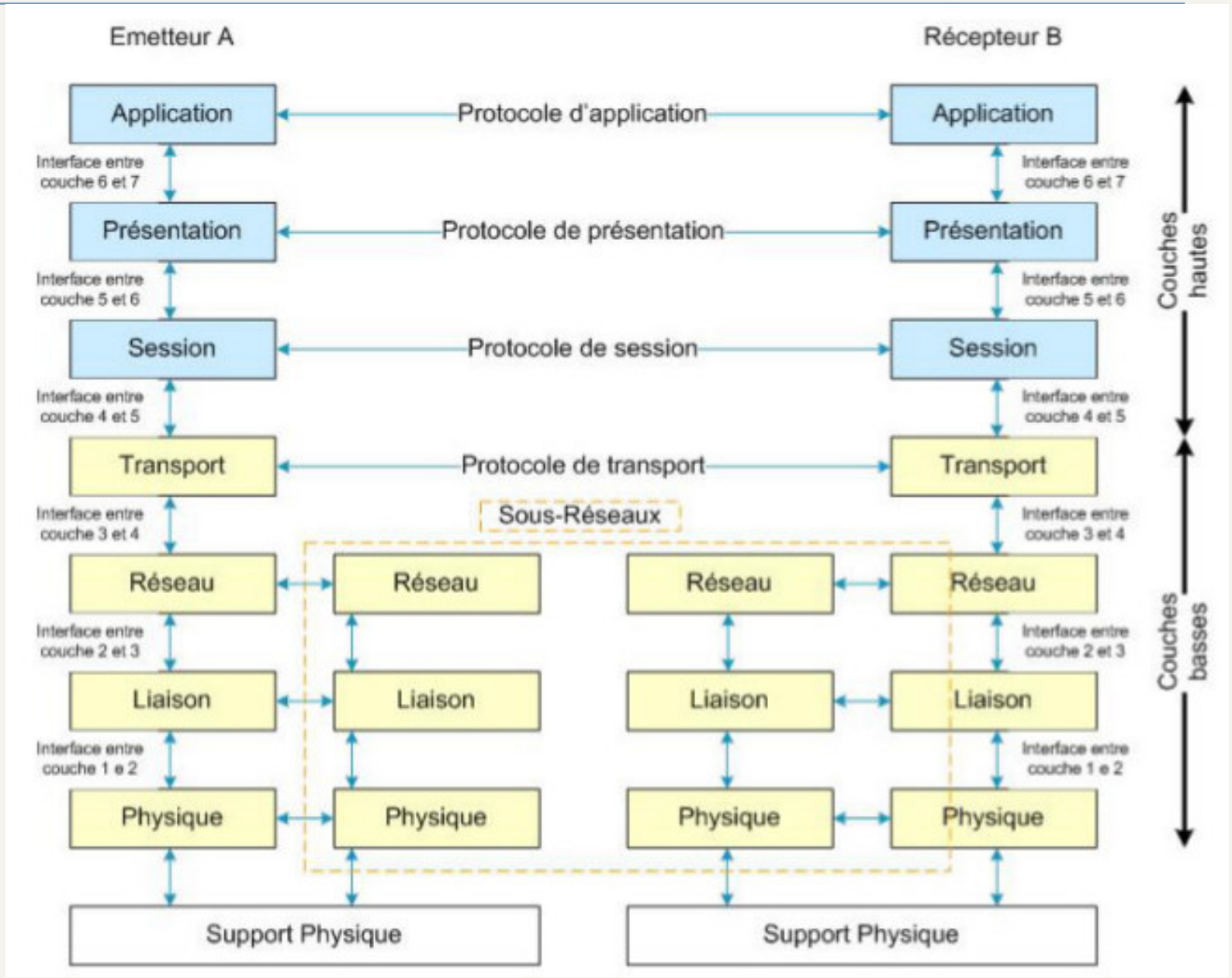


Fig 1.5 - Schéma du modèle en couche OSI

3 - Le modèle de référence OSI

- Le modèle OSI (*Open Systems Interconnection*) est un cadre conceptuel développé par l'Organisation internationale de normalisation (ISO) pour standardiser la communication entre systèmes informatiques en réseau.

- Créé dans les années 1970 et publié en 1984, ce modèle divise le processus de communication en **sept couches distinctes**, chacune ayant des fonctions spécifiques et interagissant avec les couches adjacentes.

- Cette division facilite la compréhension, la conception et le dépannage des réseaux complexes.

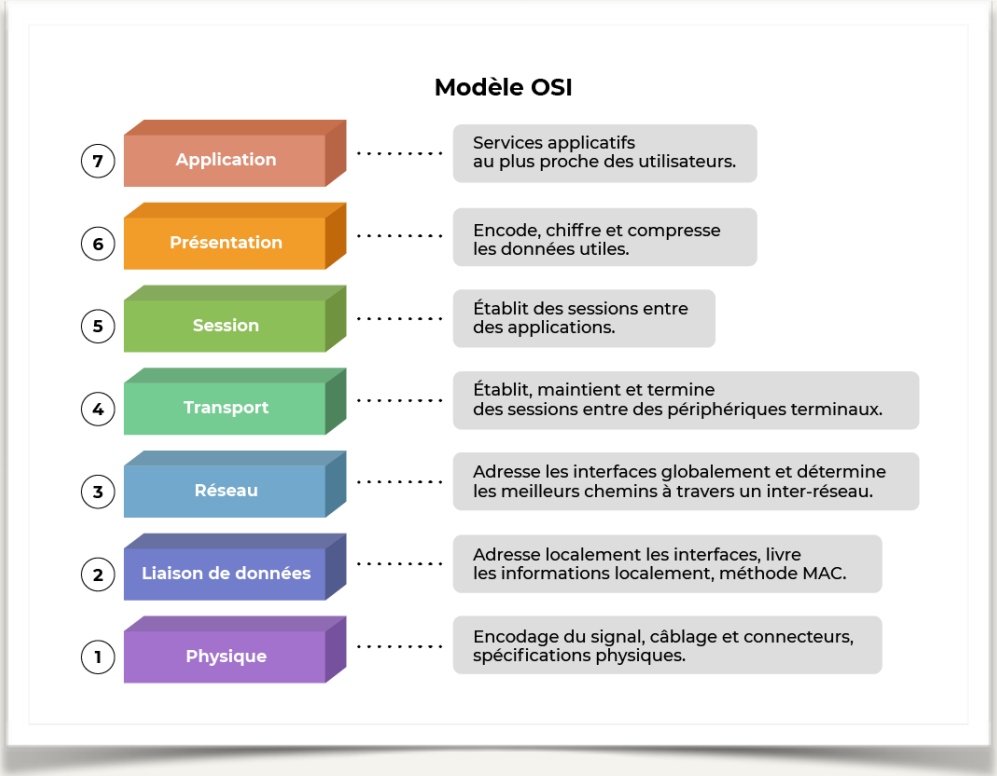
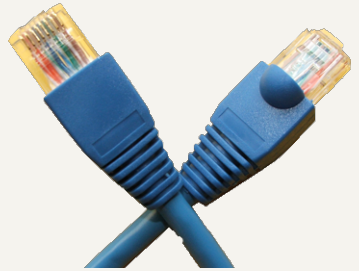


Fig 1.8 - Le modèle OSI

3 - Le modèle de référence OSI

Les sept couches du modèle OSI

Couche physique - Niveau physique - *Physical layer* - Couche 1



- Elle concerne le transfert de bits bruts sur un support physique, définissant les caractéristiques électriques et mécaniques des connexions.

Couche liaison de données - Niveau trame - *Data Link layer* - Couche 2

- Elle assure le transfert fiable de données entre deux nœuds adjacents, incluant la détection et la correction d'erreurs.
- [Pour un réseau à diffusion] Sous-couche MAC pour gérer et arbitrer les accès multiples au canal de transmission partagé

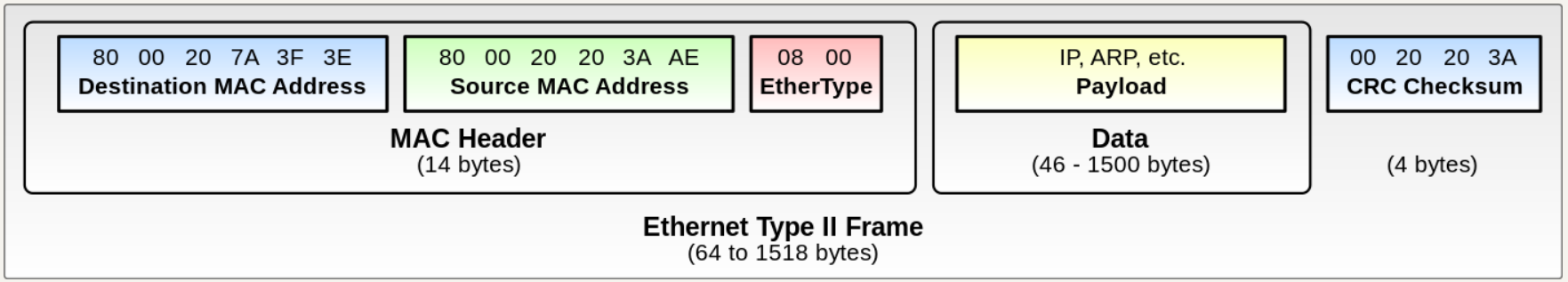


Fig 1.9 - Trame Ethernet type II

Les sept couches du modèle OSI

Couche réseau - Niveau paquet - *Network layer* - Couche 3

- Responsable du **routage** des paquets à travers le réseau, elle gère l'adressage logique et le choix des chemins.

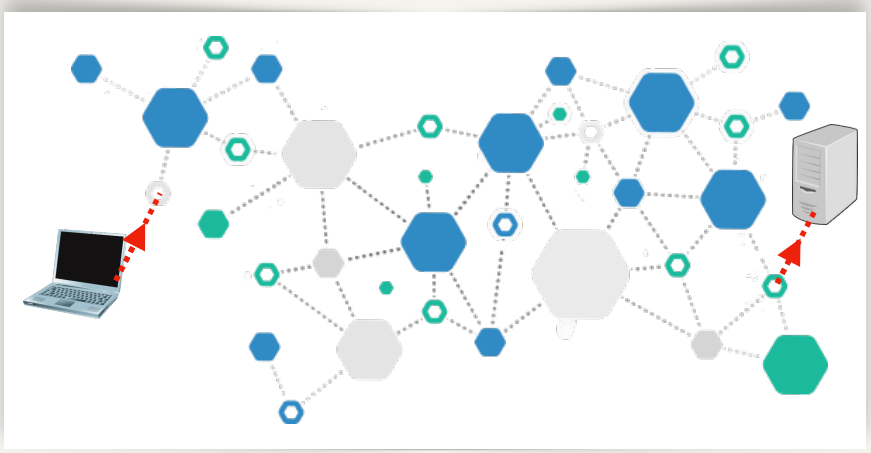


Fig 1.10 - Système de relais

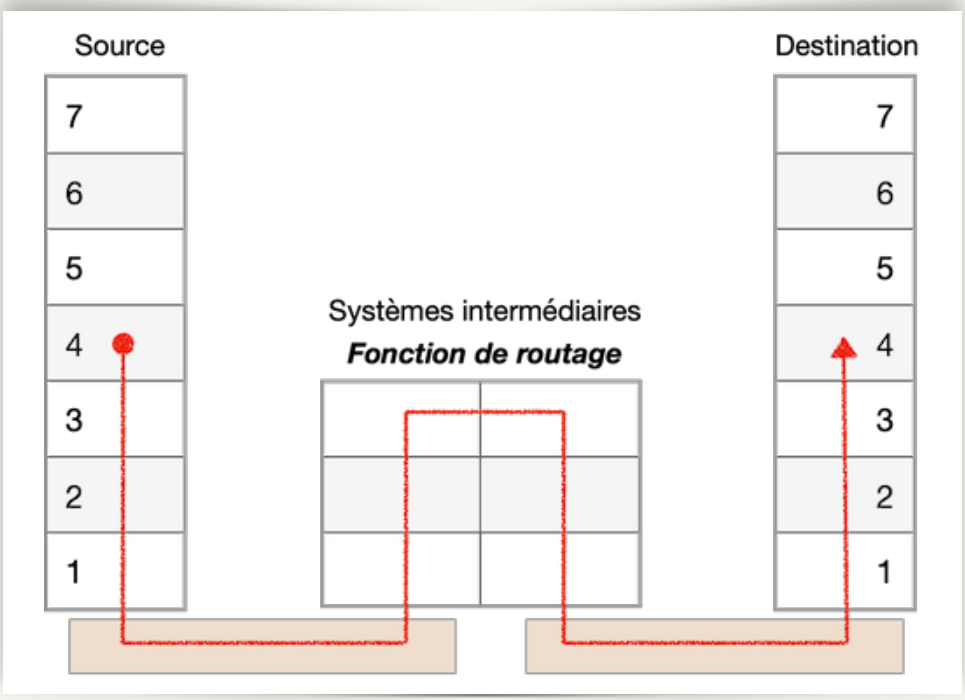


Fig 1.11 - Fonction de routage

3 - Le modèle de référence OSI

Les sept couches du modèle OSI

Couche transport - Niveau message - *Transport layer* - Couche 4

- Elle garantit un transfert de données fiable **entre systèmes finaux**, contrôlant le flux et la correction d'erreurs

Couche session - Niveau session - *Session layer* - Couche 5

- Elle gère les sessions de communication entre applications, incluant l'établissement, la gestion et la terminaison des connexions.

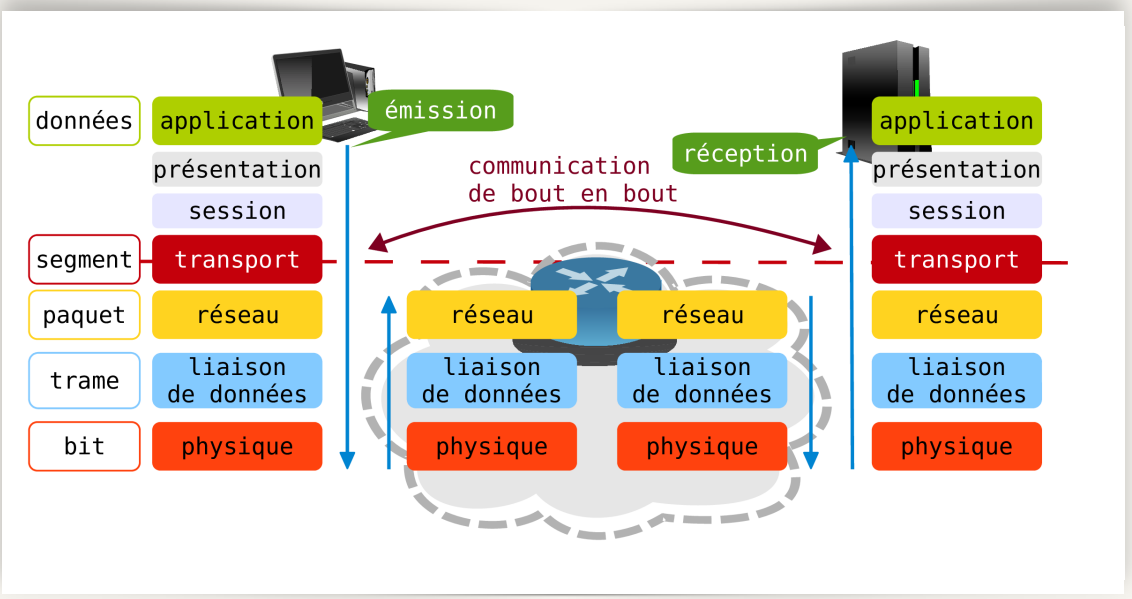


Fig 1.12 - Place de la couche transport dans le modèle OSI

3 - Le modèle de référence OSI

Les sept couches du modèle OSI

Couche présentation - Niveau présentation - *Presentation layer* - Couche 6

- Elle s'occupe de la traduction des données entre le format du réseau et celui des applications, assurant également l'encodage/décodage et la compression/décompression.

Couche application - Niveau application - *Application layer* - Couche 7

- Elle fournit des services réseau aux applications utilisateur, comme le courrier électronique ou le transfert de fichiers.

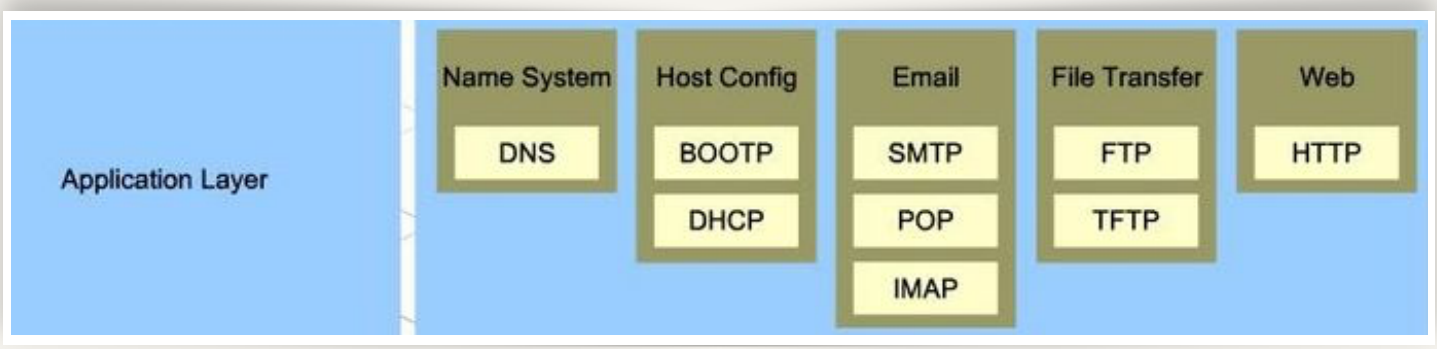


Fig 1.13 - Quelques protocoles de la couche application

4 - L'architecture TCP/IP

Internet

ARPANET - Internet

Une architecture [ou un modèle ?] à 4 couches

- Accès au sous-réseau
- Couche internet
- Couche transport
- Couche application

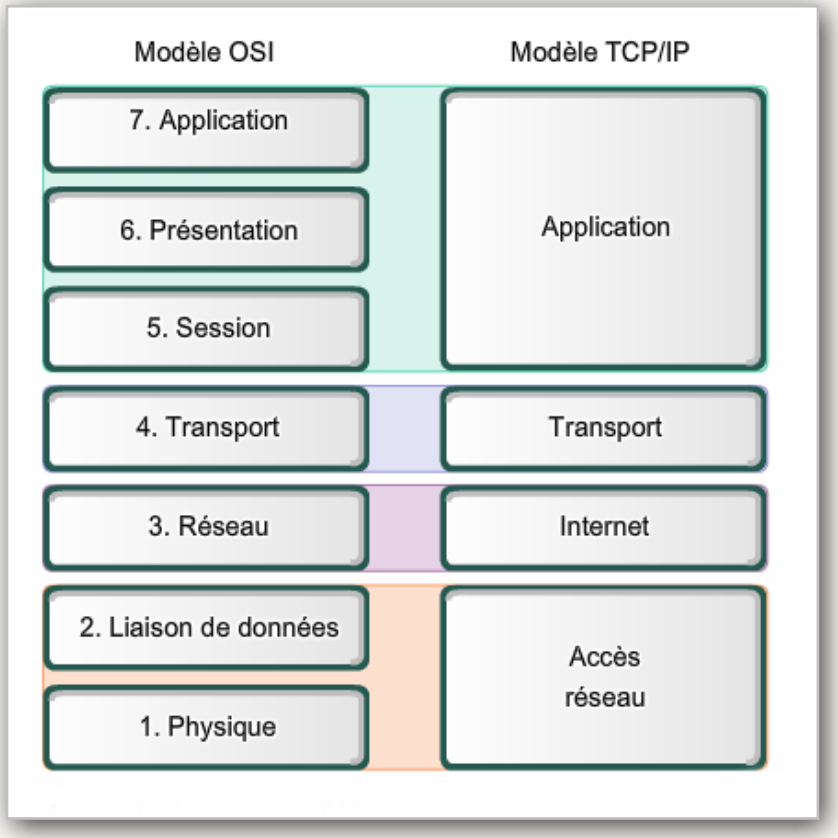
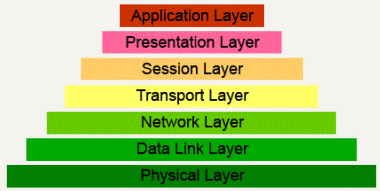


Fig 1.14 - Comparaison du modèle OSI et de l'architecture TCP/IP



4 - L'architecture TCP/IP

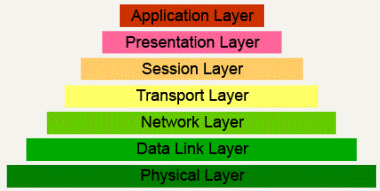
Les couches de l'architecture TCP/IP

Accès au sous-réseau (ou couche hôte-réseau)

- Non spécifiée par l'architecture TCP/IP
- Doit permettre le transfert de paquets entre hôte et routeur et entre routeurs
- Cette couche est liée au type de réseau utilisé (Ethernet, Wi-Fi, ATM…)

Couche internet

- Au niveau de la couche réseau d'OSI, son objectif est de permettre :
  - l'injection de paquets nommés **datagrammes** dans n'importe quel réseau
  - l'acheminement de ces datagrammes indépendamment les uns des autres jusqu'à destination
- Le protocole IP (*Internet Protocol*) est non fiable, sans connexion
- La couche internet assure le routage des datagrammes et la gestion des congestions



Les couches de l'architecture TCP/IP

Couche transport

- Au niveau de la couche transport d'OSI, elle définit principalement deux protocoles de bout en bout pour permettre le dialogue entre deux entités paires.
  - **TCP** : Transmission Control Protocol
  - **UDP** : User Datagram Protocol

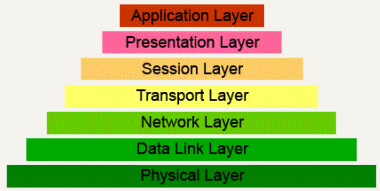
- TCP : **Transmission Control Protocol**, est un protocole fiable orienté connexion. Il permet de remettre sans erreur un flux d'octets d'un hôte à un autre en le fragmentant en segments qui sont encapsulés par le protocole IP.
- UDP : **User Datagram Protocol**, est un protocole plus simple, non fiable et sans connexion, utile lorsque ni contrôle de flux, ni ordonnancement de données n'est nécessaire.

4 - L'architecture TCP/IP

Les couches de l'architecture TCP/IP

Couche application

- Cette couche regroupe les niveaux session, présentation et application du modèle OSI. Elle contient les protocoles de haut-niveau utilisés par les logiciels pour leur besoin de communication.
- Exemples de protocoles :
  - **Transfert de fichiers** : FTP, *File Transfer Protocol ;* SFTP, *Secure File Transfer Protocol…*
  - **Messagerie électronique** : SMTP, *Simple Mail Transfer Protocol* ; POP3, *Post Office Prot.*; IMAP, *Internet Message Access Prot.* ; MIME, *Multipurpose Internet Mail Extensions…*
  - **Messagerie instantanée** : XMPP, *Extensible Messaging and Presence Protocol,* alias *Jabber…*
  - **Travaux à distance** : Telnet ; ssh, *Secure shell*
  - **Consultation et gestion d'annuaires : LDAP,** *Lightweight Directory Access Protocol*
  - **Standards et outils du web :** HTTP, *HyperText Transfer Prot.* ; URI, *Uniform Resource Identifier* ; HTML, *HyperText Markup Language* ; CGI, *Common Gateway Interface…*
  - **Traduction de nom de domaine en adresse IP** : DNS, *Domain Name System*
  - **Autres** : NFS, *Network File System ;* SNMP, *Simple Network Management Protocol ;* DHCP, *Dynamic Host Configuration Protocol ; etc.*



4 - L'architecture TCP/IP

La couche application

Exemple : Requête d'une page web

- L'internaute clique sur un lien http://example.com pour afficher une page web
- De façon détaillée, **qu'est-ce qu'il se passe ?**

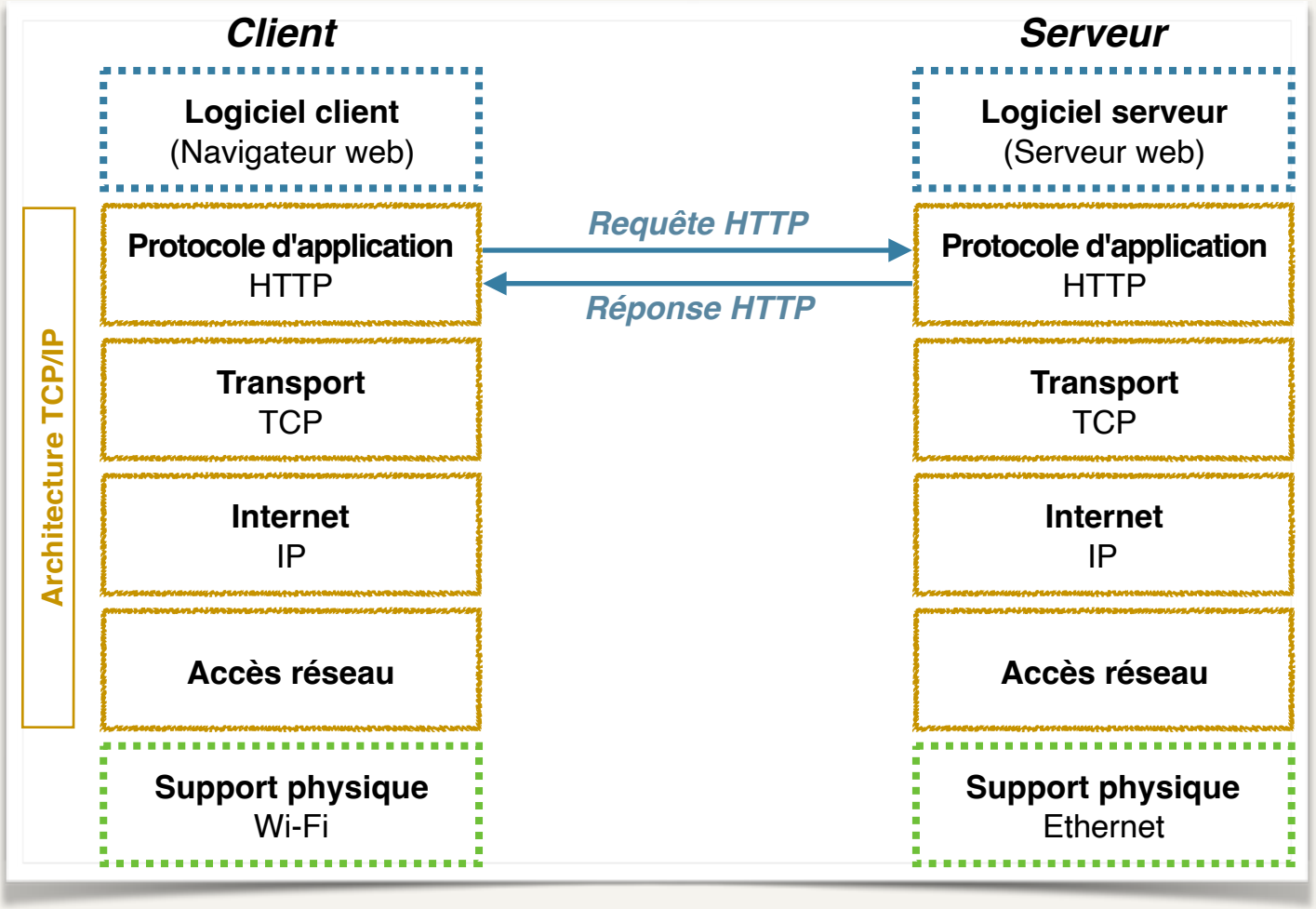


Fig 1.15 - Requête d'une page web

La couche application

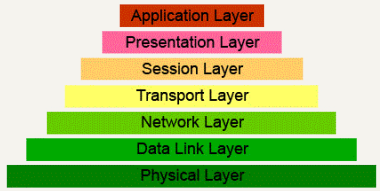
Exemple : Requête d'une page web (suite…)

- Le navigateur web
  - analyse le lien ;
  - détermine l'adresse IP du serveur via DNS : *qui est example.com ? 93.184.216.34 ;*
  - utilise HTTP ;
- HTTP (*HyperText Transfer Protocol)* envoie une requête pour obtenir une ressource sur le serveur web *example.com* : *GET / HTTP/1.1…*
  - Il utilise TCP pour le transport…
  - d'une requête GET ;
- TCP (*Transmission Control Protocol*) envoie un segment pour *93.184.216.34:80*
  - en utilisant IP (*Internet Protocol*) ;

La couche application

Exemple : Requête d'une page web (suite…)

- IP envoie un datagramme vers son routeur, pour *93.184.216.34*
  - son adresse IP source et l'adresse IP *93.184.216.34* du serveur sont spécifiés dans l'en-tête du datagramme ;
  - Il utilise la couche d'accès au réseau ;
- Qui créée les trames Wi-Fi nécessaires ;
- Qui seront codés par l'émetteur Wi-Fi pour les transmettre au routeur Wi-Fi. Le routeur trouve la route vers le serveur web et achemine le datagramme ;
- Le dernier routeur utilise Ethernet pour transmettre les trames du datagramme au serveur web *93.184.216.34*



La couche application

Exemple : Requête d'une page web (suite…)

- Sur le serveur, **trames**, puis **datagramme** IP, puis **segment** TCP sont décapsulés pour fournir au processus HTTP la requête ***GET / HTTP/1.1…***
- HTTP
  - analyse la requête,
  - recherche la ressource
  - l'inclue dans la réponse HTTP : ***HTTP/1.1 200 OK…***
- Cette réponse sera transmise au client via TCP, **avec toutes les étapes inverses**
- HTTP du client obtient la réponse via TCP
- Le navigateur web
  - interprète cette réponse
  - l'affiche s'il dispose de toutes les ressources nécessaires
  - (sinon, il sollicite HTTP pour obtenir les ressources qu'il n'a pas en cache)

- Voir aussi : https://www.wireshark.org/

Contenu du chapitre

- Les autoroutes de l'information : nids de poules et travaux en tous genres **(couche physique)**
  - Concepts et problèmes de la transmission de données : erreurs de transmission, le contrôle d'erreur, notion de bande passante, traitement des signaux, atténuation, modulation, multiplexage, commutation, synchronisation d'horloge, problèmes de caractère et de bit stuffing.

Rappel : rôle de la couche physique du modèle OSI

Couche physique - Niveau physique - ***Physical layer***

- Transfert de **train de bits** d'information sur le support physique
- unité de PDU : bit

- Définition des **supports physiques** et des moyens d'y accéder

- Spécifications des interfaces :
  - **mécaniques** (définition, dimension des connecteurs)
  - **électriques**
  - **fonctionnelles**

- Moyens d'adaptation
  - Transformation de trains de bits en **signaux** adaptés au support, et vice-versa.

1 - Préambule

Le contrôle physique du circuit de données

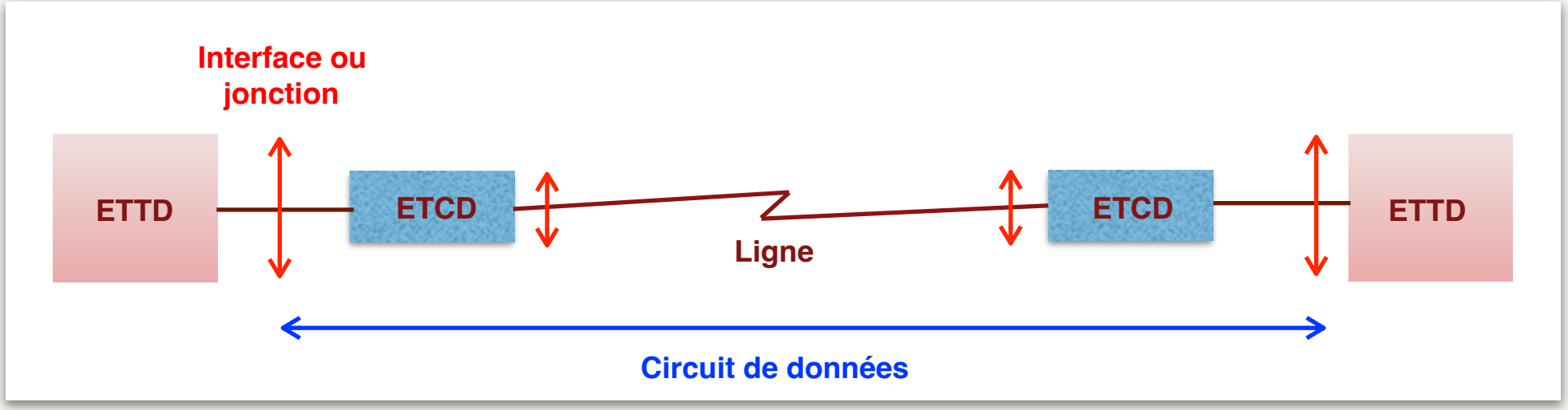


Fig 2.1 - Le circuit de données

ETTD : Équipement Terminal de Traitement de données

- DTE : *Data terminal equipment*
- Ex. : **Ordinateur**, terminal, imprimante…

ETCD : Équipement Terminal de Circuit de données

- DCE : *Data circuit-terminating equipment*
- Ex. : **Modem**
- L'ETCD gère la liaison de la ligne à chaque extrémité et adapte le signal binaire entre l'ETTD et la ligne de transmission via un codage et/ou une modulation

Définitions

Information analogique

- Liée à la variation continue d'un phénomène physique

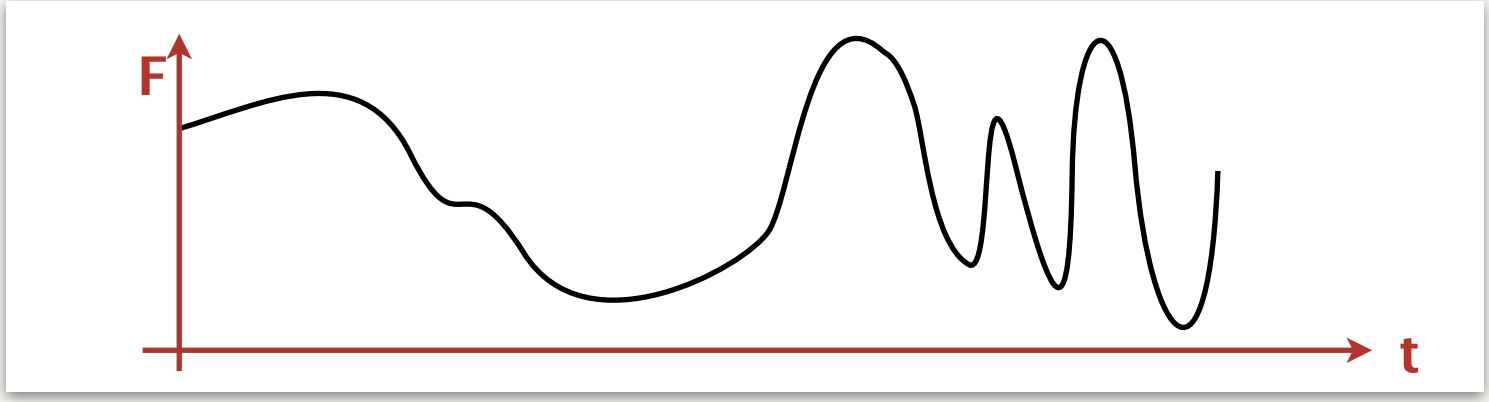


Fig 2.2 - variation continu de F en fonction du temps

Information numérique

- Elle résulte :
  - d'une source discontinue (ou discrète) ;
    - où les variables ne peuvent avoir qu'un nombre fini de valeurs
  - de l'assemblage d'éléments indépendants (alphabet...)
  - de la numérisation d'une information analogique
    - (=> échantillonnage, quantification et codage)

Définitions

Bit

- Unité d'information
- `0 | 1`

- Les **supports physiques** véhiculent des signaux qui représentent les informations transmises
  - Support métallique => signaux électriques
  - Ondes radio
  - Infrarouge
  - Fibre optique => Ondes optiques

- La transmission sur le support passe par des technique de
  - **codage en bande de base**
  - ou de **modulation**

Définitions

Débit binaire *(bitrate* ou *bit rate)*

- Nombre de bits par seconde émis sur le support de transmission
- Quantité de données transmises dans un intervalle de temps fixé

$$D = \frac{V}{T}$$

D : débit binaire en bit/s
V : Volume d'information en bit
T : durée d'émission

Fig 2.3 - Débit binaire

Un débit s'exprime en **bit/s** (ou ses multiples : kbit/s ; Mbit/s ; Gbit/s…)

- ⚠ **Éviter** :
  - *bps*
  - octet/s ; 1 ko/s… (mais concevable pour la **couche application**)

Définitions

Latence *(lag)*

- Temps de transit entre l'émission d'un bit et sa réception
- Cela inclus le temps de propagation sur les supports et les temps de traitement par les équipements actifs du réseau

Gigue *(jitter)*

- **Variation** de la latence dans le temps

Définitions

Délai d'attente aller-retour (***RTT***, *round-trip time* ou ***RTD***, *round-trip delay time*)

- Temps que met un signal pour parcourir l'ensemble d'un circuit fermé.
- la commande `ping` mesure, entre autres, le délai entre l'envoi d'une requête ICMP '*echo-request*' et la réponse '*réponse echo*'. Exemple :

```
$ ping -c3 example.com
PING example.com (93.184.216.34): 56 data bytes
64 bytes from 93.184.216.34: icmp_seq=0 ttl=56 time=86.834 ms
64 bytes from 93.184.216.34: icmp_seq=1 ttl=56 time=86.859 ms
64 bytes from 93.184.216.34: icmp_seq=2 ttl=56 time=86.612 ms
--- example.com ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 86.612/86.768/86.859/0.111 ms
```

Définitions

Codage

- Faire correspondre à chaque symbole d'un alphabet une représentation binaire (un mot-code)
- L'ensemble des mots-code = **le code**
- Exemple :
  - Code ASCII (voir ascii-table.pdf) ;
  - Unicode

@	64	40	At sign
A	65	41	Capital A
B	66	42	Capital B
C	67	43	Capital C
D	68	44	Capital D
E	69	45	Capital E
F	70	46	Capital F
G	71	47	Capital G
H	72	48	Capital H
I	73	49	Capital I
J	74	4A	Capital J

Fig 2.4 - Extrait du code ASCII

Définitions

Bande passante *(bandwidth)*

- C'est une caractéristique physique d'un **support de transmission**, mesurée en hertz (Hz)
- Largeur de la bande de fréquences des signaux qui sont transmis correctement (sans affaiblissement dommageable)
- Pour transmettre, **il faut de la bande passante !**
- => Éviter ou réduire :
  - les bruits (dûs aux champs électro-magnétique, …),
  - les interférences (ondes radio, …)
  - les atténuations (longueur et nature du support, …)
- La bande passante est généralement définie à -3 dB, d'où une atténuation en puissance de moitié.
  - Exemple : la bande passante d'une paire téléphonique du **Réseau Téléphonique Commuté** est de 300 - 3 400 Hz.

La **largeur de bande** (*Spectral width*) est une caractéristique du **signal.**

- On étudie le spectre du signal pour déterminer sa largeur de bande.
- La bande passante du support doit être plus large que la largeur de bande pour éviter une déformation lors de la transmission.

Adaptation du signal à transmettre

Transcodage ou codage en ligne

- Un signal binaire ne peut être transmit sans adaptation sur une ligne de transmission

Fig 2.4. - Codage en ligne

Types de codage en ligne

- Codage en **bande de base**
- Codage **complets**
- **Modulation** : transmission en **large bande**

Challenge

- Obtenir plus de débits, avec peu d'énergie.
- En associant différents types de codage.

La modulation

Une porteuse à haute fréquence (HF) est modulée par le signal de données.

Trois types de modulations :

- Modulation d'amplitude
- Modulation de fréquence
- Modulation de phase

Le codage à transmettre subit une translation de fréquence, autour de la fréquence centrale

- Réduction de la dispersion d'harmonique

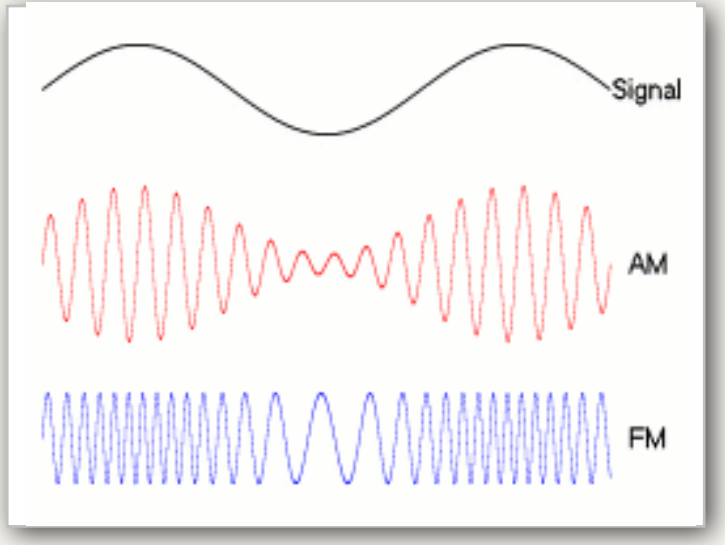


Fig 2.5 - Modulation AM et FM

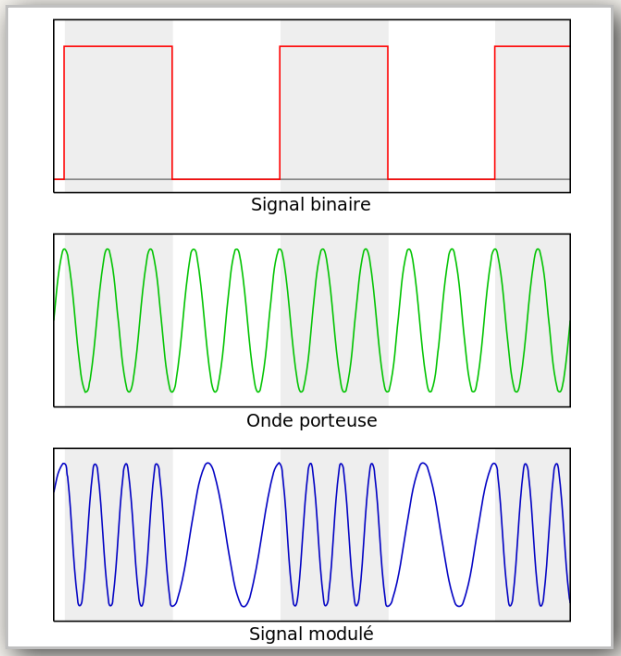


Fig 2.6 - Modulation de fréquence

La modulation

Modulation d'amplitude

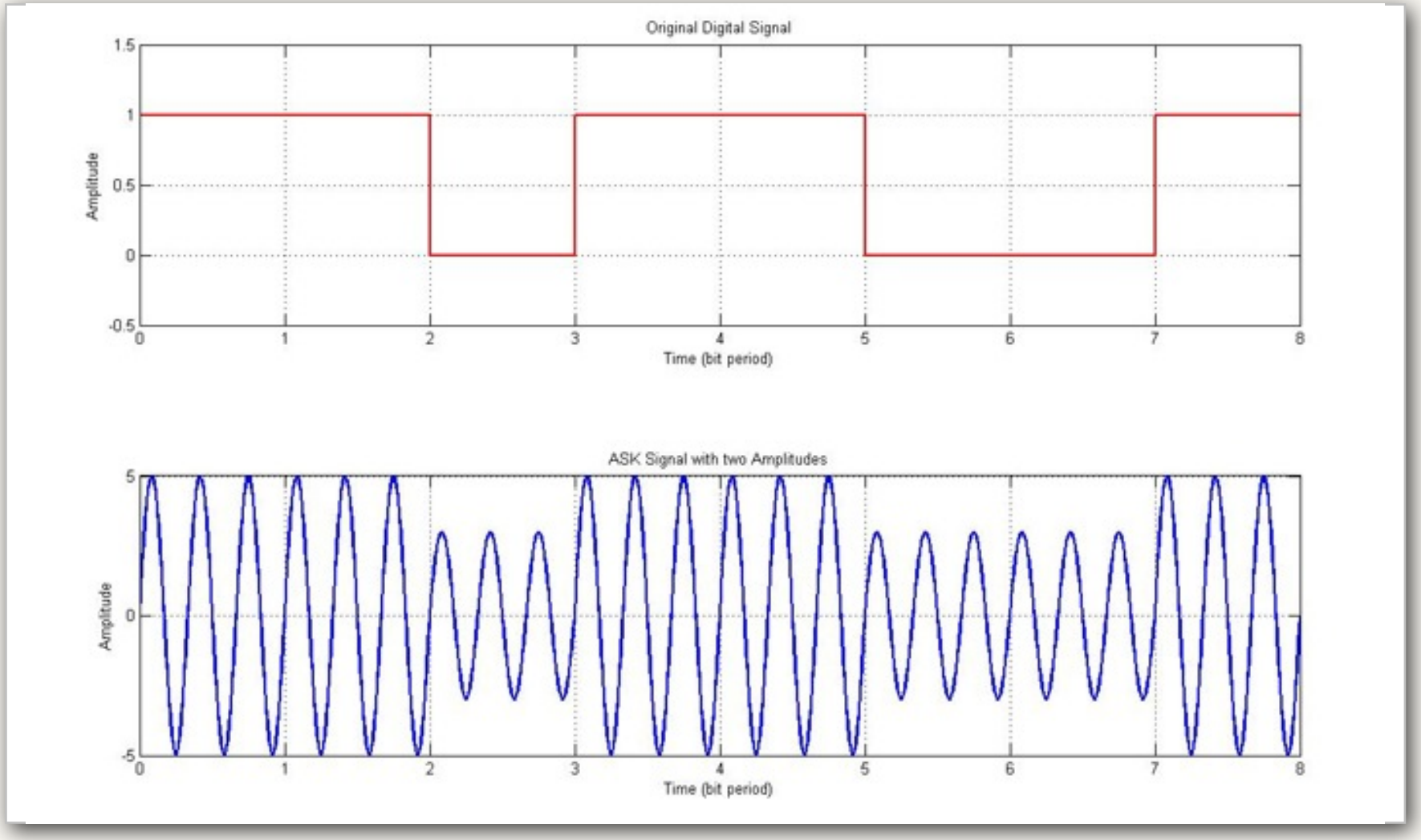


Fig 2.7 - Modulation d'amplitude

La modulation

Modulation de fréquence

- F*requency-shift keying* (FSK)
- Modulation par déplacement de fréquence (MDF)

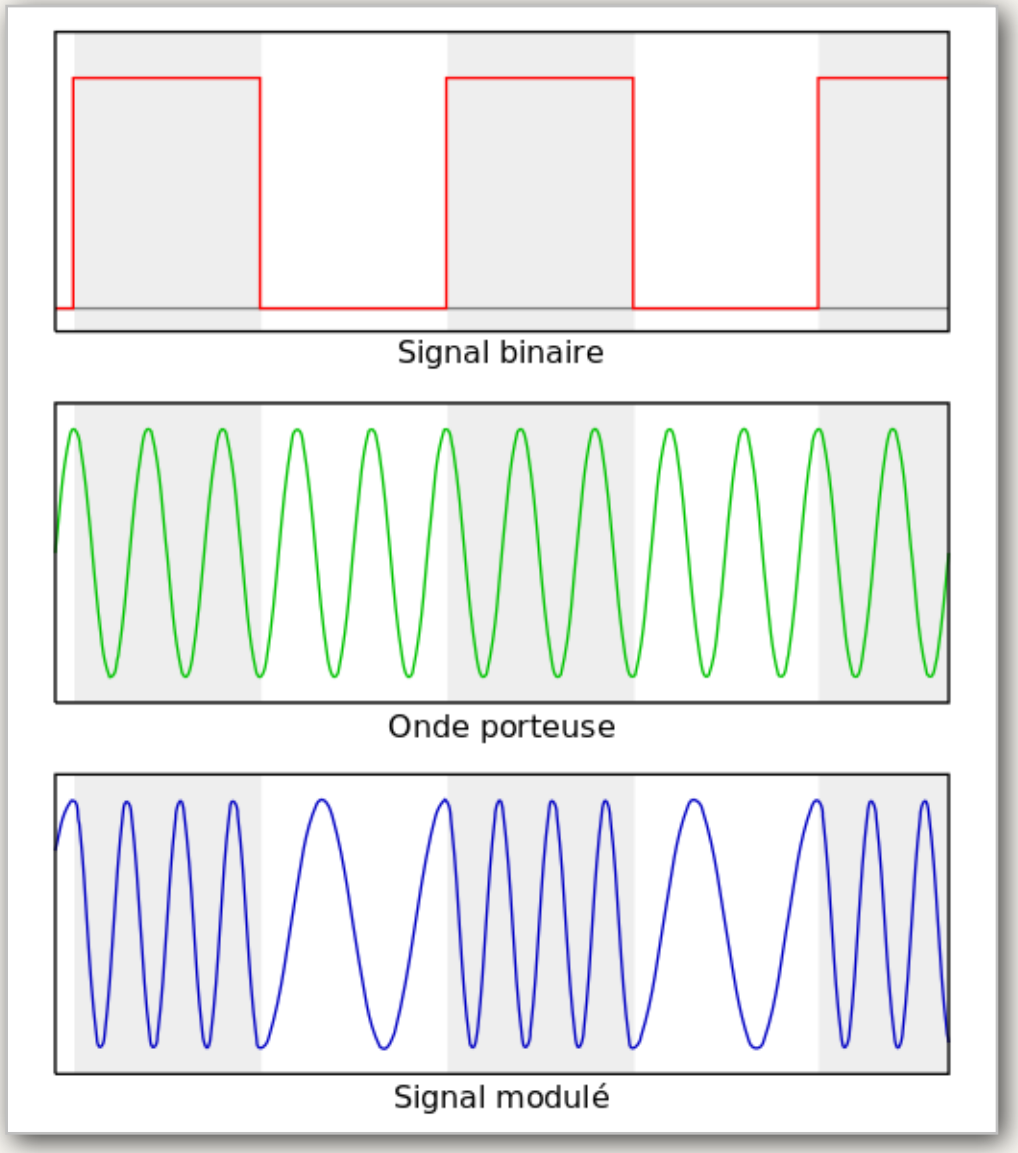


Fig 2.8 - Modulation de fréquence

La modulation

Modulation d'amplitude en quadrature

- Modulation d'**amplitude** et de **phase**
- *Quadrature Amplitude Modulation* (QAM)
- Voir : https://claude-gimenes.fr/fr/p/21/509/2957

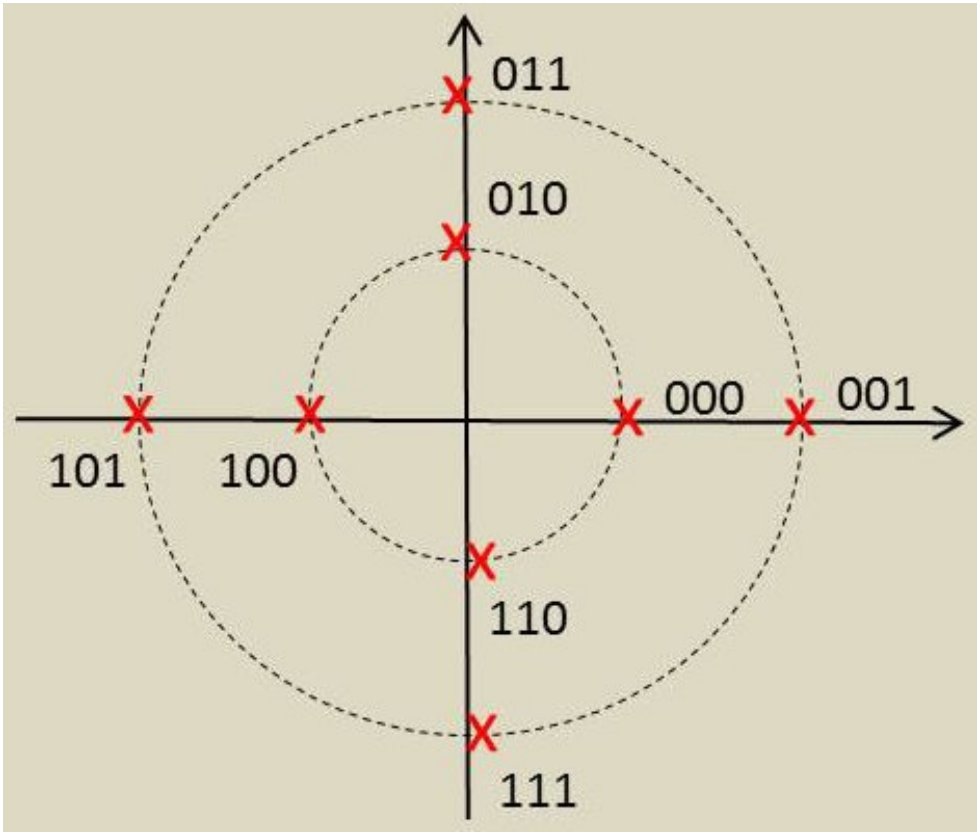


Diagramme de constellation
QAM à 8 états

Fig 2.9 - Modulation d'amplitude en quadrature

La modulation

Modem ; transmission en large bande

- L'émetteur réalise la **modulation** avec un ***modulateur.***
- Le récepteur utilise un ***démodulateur*** pour restituer les données d'origine.
- Dans un ETCD (Équipement Terminal de Circuit de données), ces deux fonctions sont réalisées par un ***modem.***
- La transmission de signaux modulés est appelée **transmission en large bande.**

Quantifier le débit en fonction de la bande passante

Théorème de Shannon ou d'échantillonnage :

- La fréquence d'échantillonnage $F_{éch}$ doit être en rapport avec F_{max}, la plus grande fréquence du signal à convertir :
- $F_{éch} \geq 2 F_{max}$

Rapport signal / bruit = Signal / Noise = **S/N**

- Rapport entre la puissance **S** du signal transmis et la puissance **N** du bruit, souvent exprimé en décibel (dB) :
- $S/N_{db} = 10 \log_{10} (P_S / P_N)$
  - Exemple : $P_S / P_N = 100$ => $S/N_{db} = 20$ dB

Formule de Shannon : débit en fonction du bruit. Cela permet d'évaluer une **capacité de transmission** C d'un canal bruité :

- $C = W \log_2 (1 + S/N)$
  - C : Débit max. ; capacité de transmission, en **bit/s**
  - W : Bande passante du canal de transmission, en **Hz**
  - S/N : Rapport signal sur bruit (en valeur — pas en dB)

Quantifier le débit en fonction de la bande passante

Rapidité de modulation (ou rapidité de transmission)

- La rapidité de modulation R, exprimé en bauds (bd), mesure le nombre de signaux transmits par unité de temps.
- $D = R \log_2 (V)$
  - D : débit binaire, en bit/s
  - R : rapidité de modulation en bd (baud)
  - V : valence du signal ; nombre d'états significatifs possibles du signal
    - Pour un signal bivalent,V=2 et D = R
    - Pour un signal quadrivalent, V=4 et D = 2 R

Formule de Nyquist

- $D_{max} = 2 W \log_2 (V)$
  - D_{max} : Débit max., en **bit/s**
  - W : Bande passante du canal de transmission, en **Hz**
  - V : Valence ; correspond au nombre de niveaux significatifs d'un signal.

Définitions

- La commutation est l'action d'associer temporairement des voies de transmission ou des circuits de télécommunication pendant la durée nécessaire au transfert de l'information.
- Les types de commutation
  - **Commutation de circuits** (*circuit switching*)
    - Établir un circuit de bout-en-bout entre deux utilisateurs avant toute transmission d'informations.
    - Monopoliser ce circuit pendant toute la communication.
    - Libérer le circuit au terme de la communication.
    - **Exemple** : Pour établir une communication téléphonique entre 2 abonnés A et B, un circuit physique doit être établi à travers les différents relais du système téléphonique.

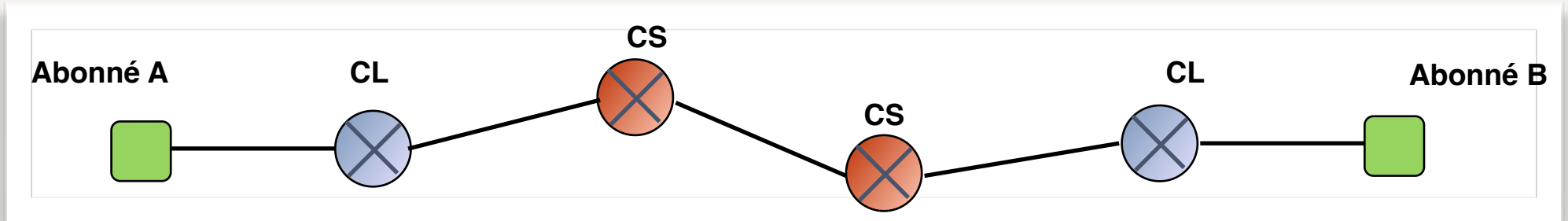


Fig 2.12 - Liaison téléphonique entre 2 abonnés

Définitions

- **Commutation de messages**
  - La source constitue un **message** à partir d'un bloc de données et le fait passer au commutateur auquel il est raccordé.
  - Le commutateur
    - **stocke** le message
    - le **vérifie**
    - trouve la route pour le **faire suivre** vers le destinataire
    - le transmet au commutateur suivant
  - Technique nommée « **Store & forward** », traduit par « **Stocker, vérifier, faire suivre** »
  - Ancêtre : le système télégraphique

Fig 2.13 - Système télégraphique

Définitions

- **Commutation de paquets** (*packet switching*)
  - C'est une commutation de messages, avec une taille **réduite** et **fixe** des messages
  - Le bloc de données est donc fragmenté par l'émetteur en paquets (ou en trames)
  - Les nœuds de transfert traitent ces paquets rapidement car :
    - stockés en RAM et non sur disque
    - algorithmes plus simples *(à cause de la taille fixe des paquets)*
    - Technique nommée « **Store & forward** »
    - Mais on traduit par : « **Stocker, vérifier, faire suivre** »
  - Nœuds de transfert : **routeurs** et **commutateurs**

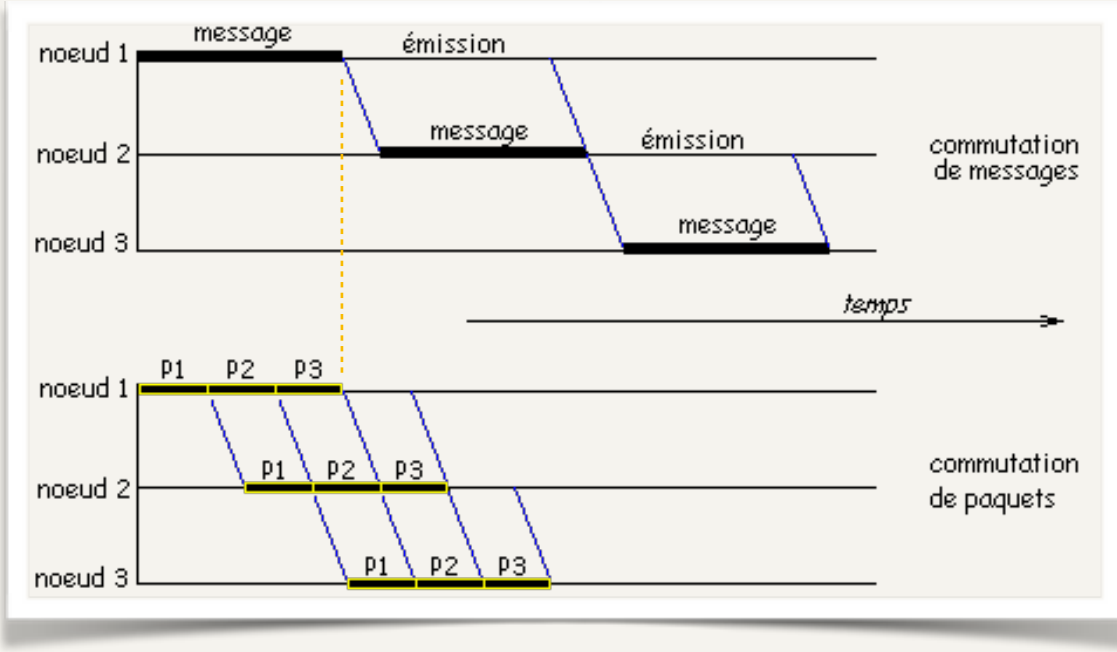
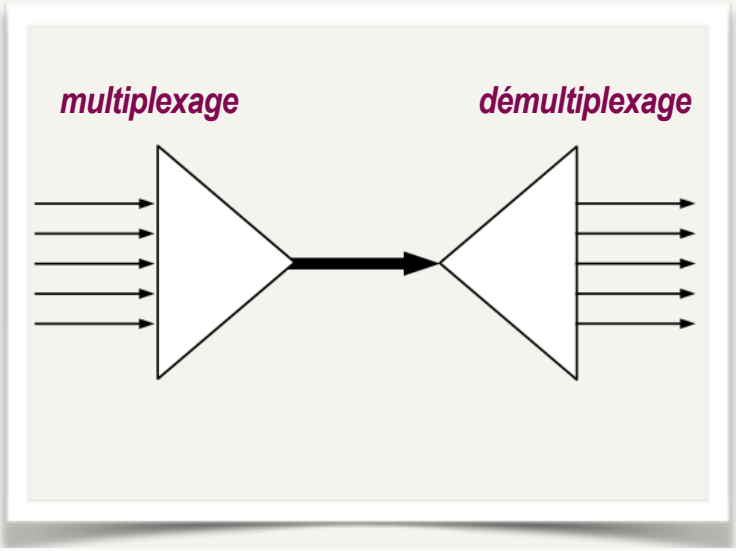


Fig 2.14 - Commutation de messages vs commutation de paquets

4 - Le multiplexage

Définitions

- Le **multiplexage** (*Multiplexing*) est une technique qui consiste à faire passer plusieurs informations dans un même canal.
  - Il existe deux techniques principales de multiplexage : fréquentiel et temporel
  - Démultiplexage : opération inverse



- **Voie composite** ou **voie haute vitesse :** canal de transmission entre 2 multiplexeurs / démultiplexeurs

  Voies basse vitesse : les voies incidentes

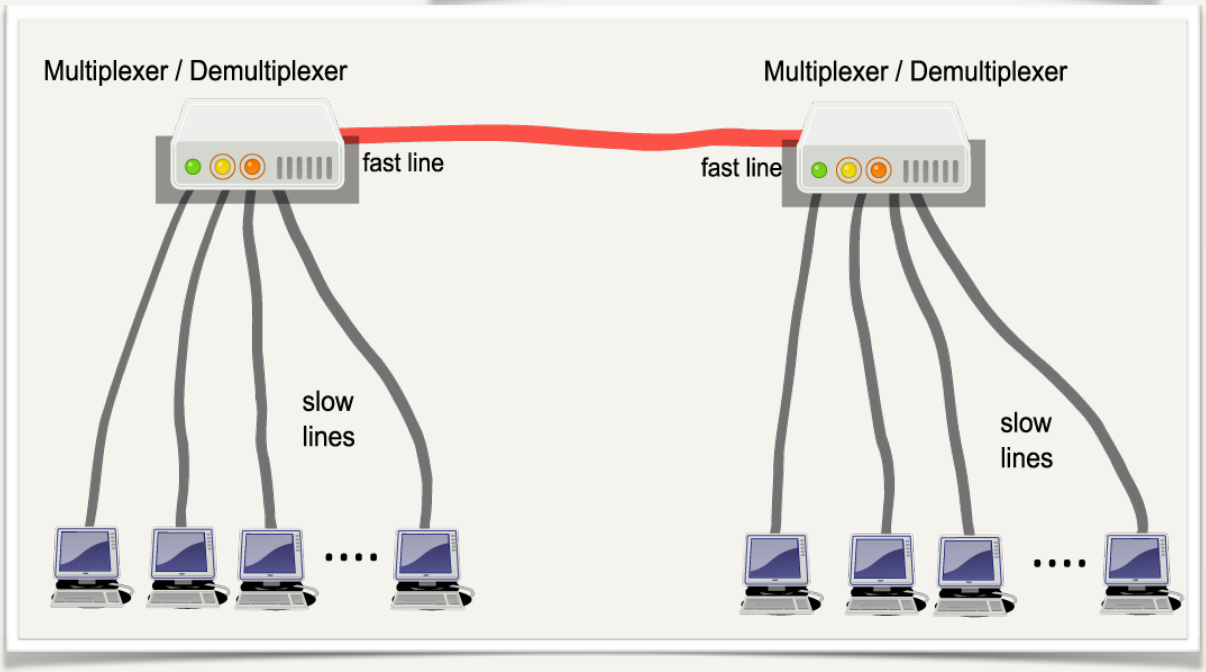


Fig 2.15. - Multiplexage

Multiplexage fréquentiel ; FDM

- **FDM** (*Frequency Division Multiplexing*)
  - MRF (Multiplexage par répartition de fréquence)
  - La bande passante de la voie composite est partagée en une série de sous-bandes, ou canaux

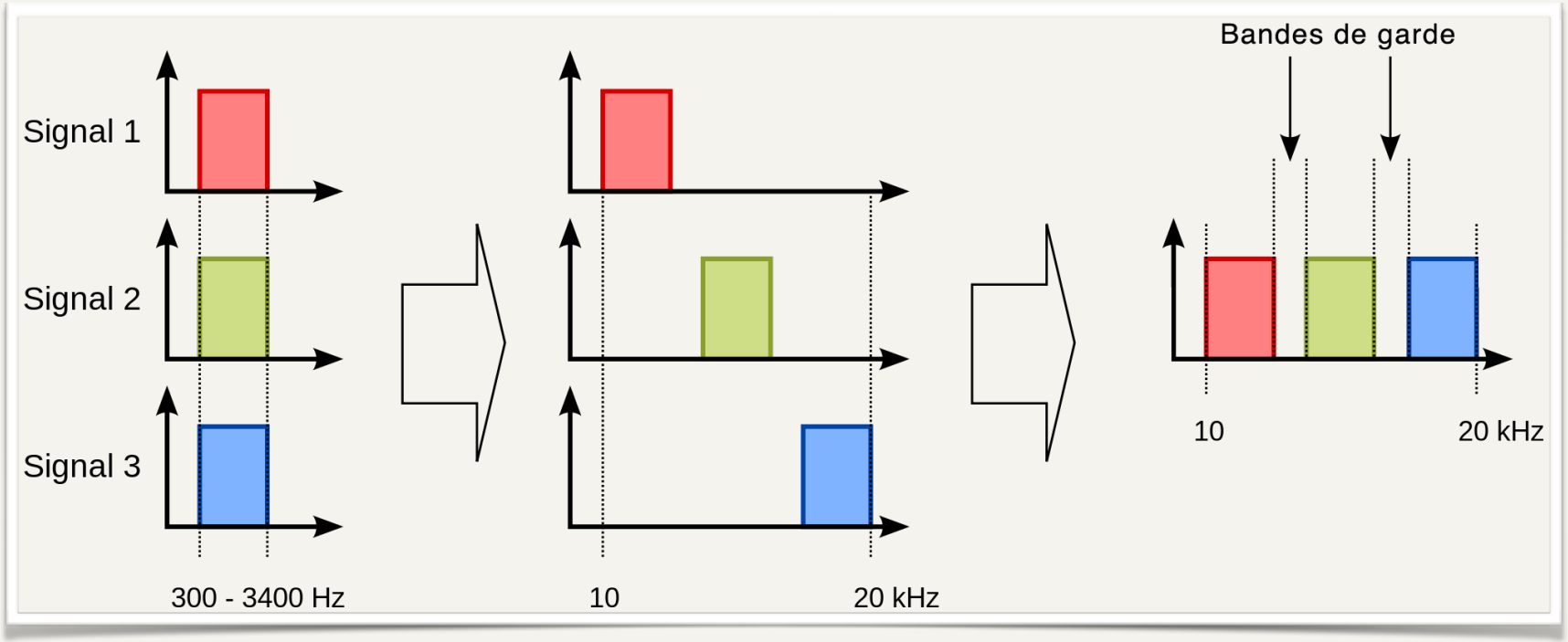


Fig 2.16. - Multiplexage fréquentiel

Multiplexage fréquentiel ; FDM

- **ADSL** (*Asymmetric Digital Subscriber Line*) combine deux techniques de modulation :
  - **FDM**
  - la modulation **DMT** (*Discrete MultiTone*)

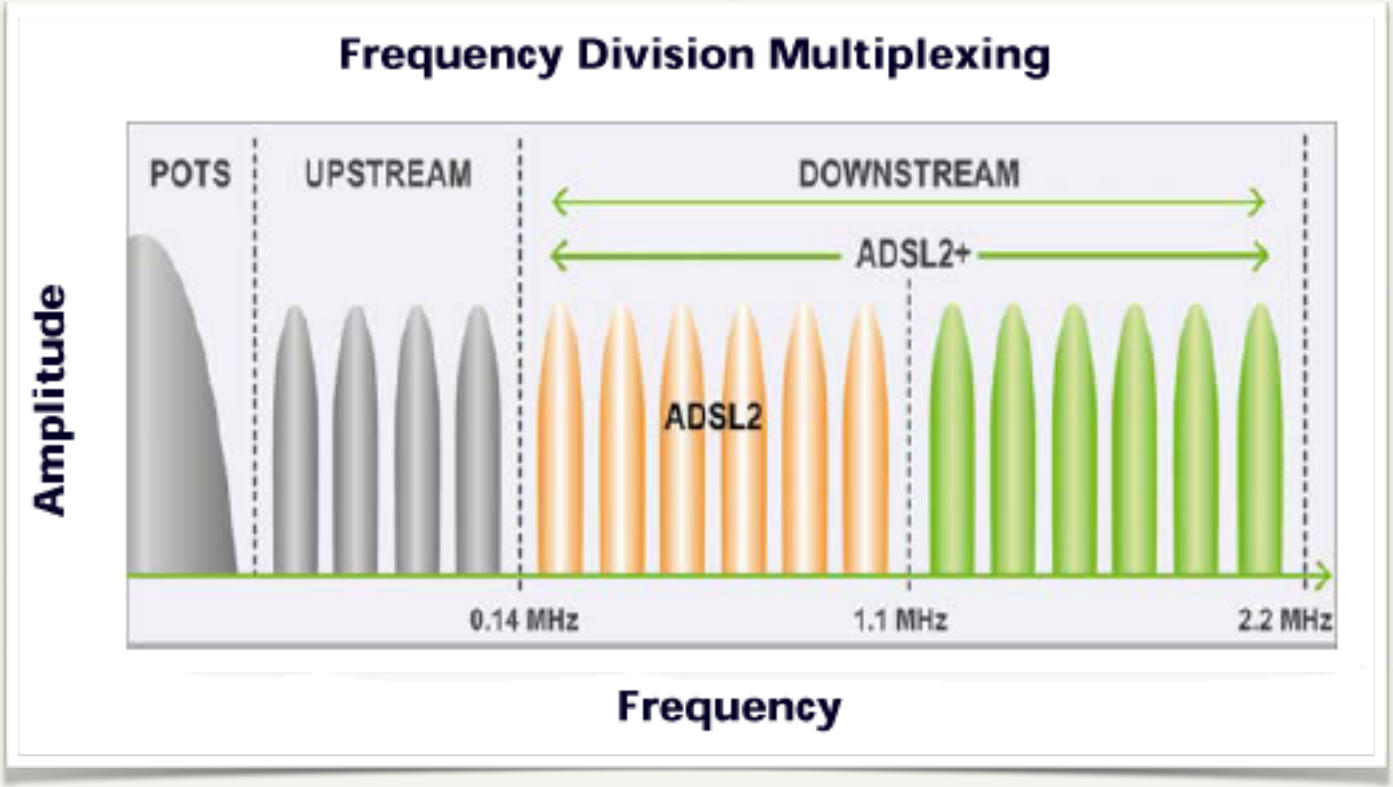


Fig 2.17. - ADSL

Multiplexage temporel ; TDM

- TDM (*Time Division Multiplexing*)
  - AMRT (Accès Multiple à répartition dans le temps)
  - Les utilisateurs émettent chacun leur tour pendant un bref intervalle de temps (IT).
  - La totalité de la bande passante de la voie composite est donc allouée à chaque utilisateur à tour de rôle.

Fig 2.18. - Multiplexage temporel

5 - Topologies

Topologie physique ou topologie logique

- Bus
- Anneau
- Arbre
- Réseau maillé
- Étoile

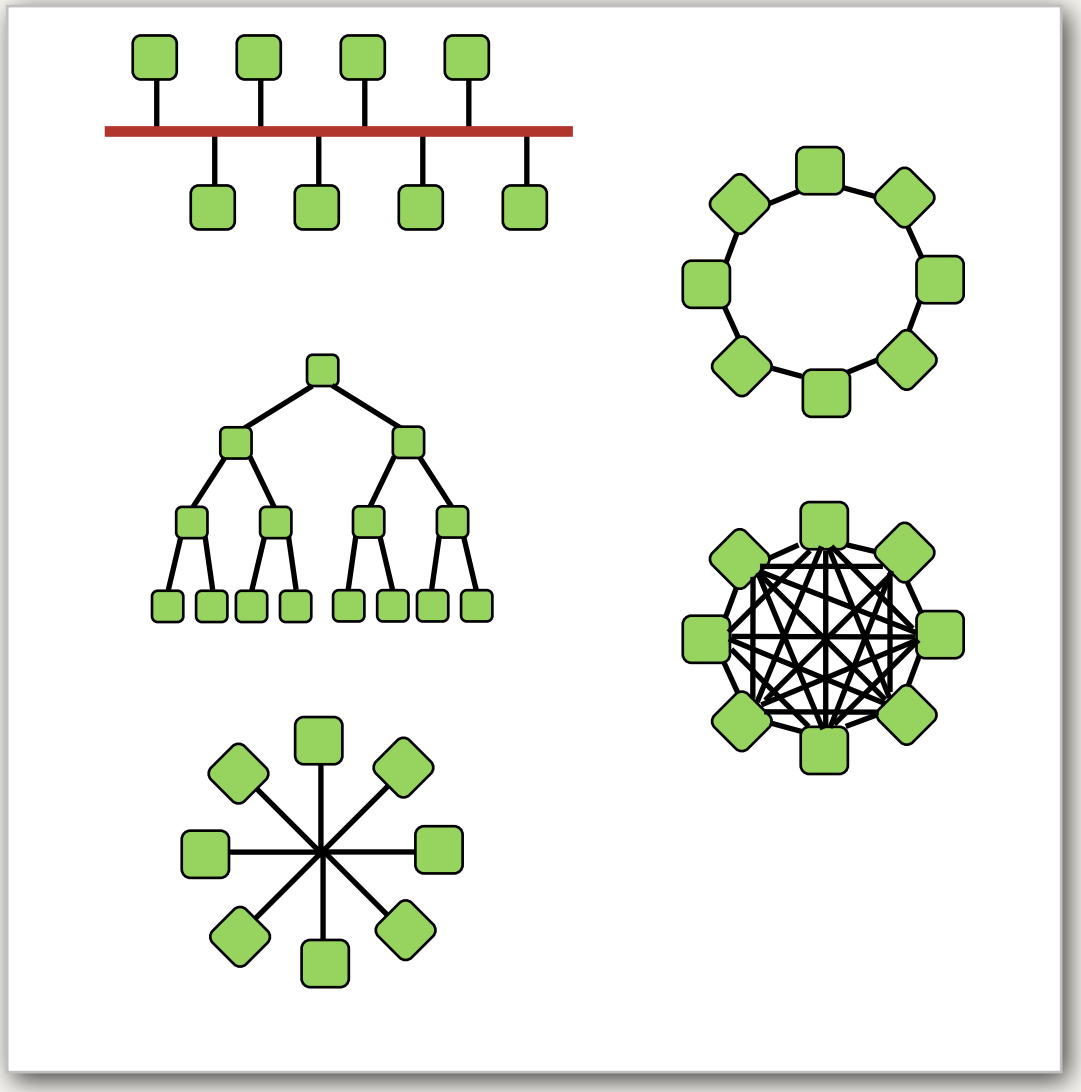


Fig 2.19 - Différentes topologies

Voir Annexe (hors cours) :

https://utc505.seancetenante.com/documents/Annexe-1-Support-physique-Telephonie.pdf

1 - Préambule

Contenu du chapitre

- *Collectivisme ou Libre entreprise… à la recherche d'un modèle équitable*
  - Grandes familles de protocoles à compétition et à coopération, détail sur CSMA/CD et CSMA/CA en mode infrastructure. Ponts et commutation.

2 - Objectif de la couche liaison de données

Rappels

Couche liaison de données - Niveau trame - *Data Link layer*

- Transfert de **trames** entre deux systèmes adjacents
- Établissement, maintient, contrôle et libération du **lien logique** entre les deux entités
- Constitution et transmission de trames **bien délimitées**
- **Détection** et correction d'erreurs
- [Pour un réseau à diffusion] **Sous-couche MAC** pour gérer et arbitrer les accès multiples au canal de transmission partagé

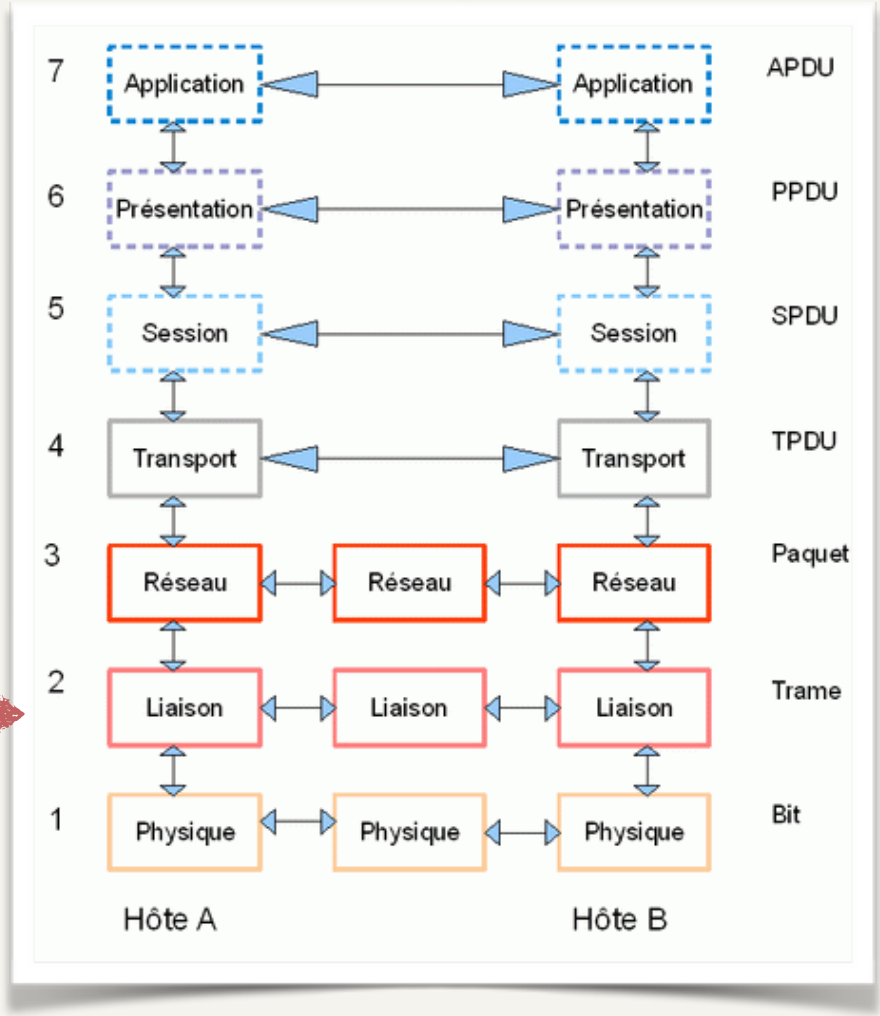


Fig 3.1 - Le modèle OSI

2 - Objectif de la couche liaison de données

Services offerts

La couche liaison de données **fournie** à la couche réseau les services nécessaires pour transmettre les paquets d'un nœud source à un nœud destination adjacent.

On distingue trois catégories de service

- **Service sans connexion et sans accusé de réception**
  - À condition que les taux d'erreur soient faibles et que les corrections ou reprises sur erreur soient assurées dans une couche supérieure
  - C'est le cas de **LAN** (*Local Area Network*), de trafic temps réel, de support fiable
- **Service sans connexion et avec accusé de réception**
  - Le récepteur doit acquitter chaque trame reçue
  - Cas de liaison peu fiable (**réseau sans fil** par ex.)
- **Service avec connexion et avec accusé de réception**
  - La liaison de données garantie que chaque trame est reçue, une fois et une seule, dans l'ordre d'émission

2 - Objectif de la couche liaison de données

Services offerts

Les service avec accusé de réception impliquent un contrôle d'erreur et une gestion des acquittements.

Les scénarios à prévoir sont décrits avec cette figure.

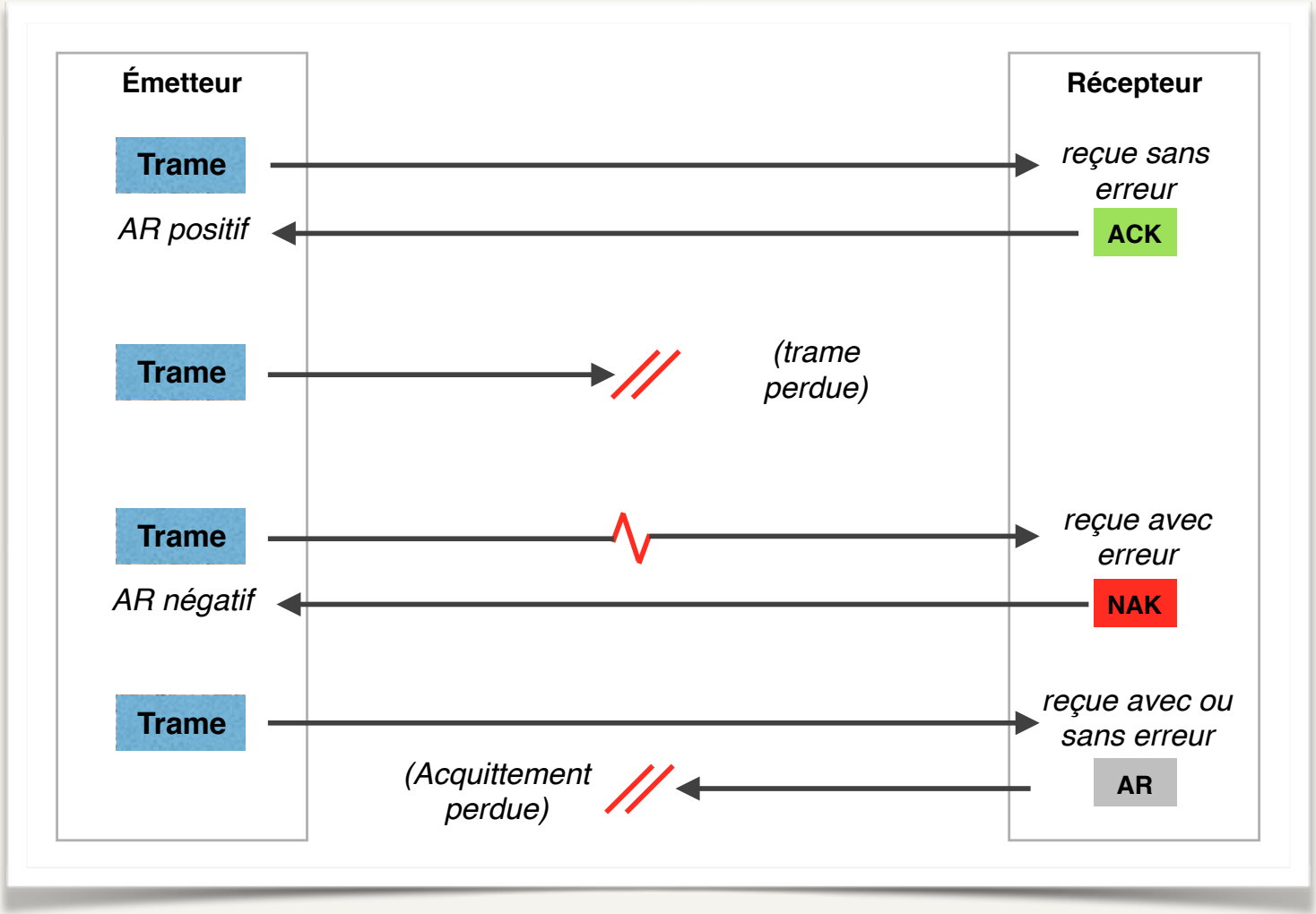


Fig 3.2 - Scénarios pour des transmissions de trames et acquittements

2 - Objectif de la couche liaison de données

Services offerts

Avec le dernier cas, l'émetteur va retransmettre la trame et le récepteur reçoit **deux trames identiques**.

Pour gérer les trames perdues, erronées ou dupliquées, on devra **numéroter les trames**.

La couche Liaison de données **utilise les services de la couche physique**.

- La couche physique transporte des trains de bits
- La couche liaison de données doit reconnaitre des trames parmi ces trains de bits et vérifier les erreurs.

2 - Objectif de la couche liaison de données

Services offerts

Pour **délimiter les trames**, on peut utiliser :

- soit des caractères spéciaux (dans le cas de transmission en **mode caractère**)
  - DLE, STX en début de trame
  - DLE, ETX en fin de trame
  - doublement d'un DLE au sein d'une trame

Début de trame		Contenu de la trame					Fin de trame	
DLE	STX	'H'	'e'	'l'	'l'	'o'	DLE	ETX

- soit une **violation de codage** sur le support physique, en utilisant une séquence spéciale et invalide pour des données qui correspond à un délimiteur de trame
- soit un système de **fanion** de début et de fin de trame :
  - Ex. avec le **fanion** « **01111110** ».
  - L 'émetteur
    - remplace au sein des trames les séquences « **11111** » par « **111110** » ;
    - ajoute un fanion « **01111110** » en début et en fin de trame.
  - Le récepteur
    - reconnait et retire les fanions « « **01111110** » ;
    - il remplace les séquences « **111110** » par « **11111** ».

3 - Détection des erreurs

Types d'erreur et type de code

Erreur par rafale :
- entre 2 bits erronés, 0 à plusieurs bits sont erronés

Erreur isolée
- 1 bit erroné ; indépendance des erreurs

Type de code
- **Code détecteur** d'erreur :
  - on ajoute juste assez d'information pour **détecter** les erreurs
  - une trame erronée sera retransmise
- **Code correcteur** d'erreur :
  - on ajoute une redondance d'information suffisante pour que le récepteur puisse **restituer les données** originales.
  - Un tel code est utilisé pour des transmission en mode simplex

3 - Détection des erreurs

Distance de Hamming

La distance de Hamming entre deux mots de code N_1 et N_2 est le nombre de bits différents :

- On calcule $N_1 \oplus N_2$
- Le nombre de bits à 1 dans $N_1 \oplus N_2$ est la distance de Hamming

La distance de Hamming d'un code complet est la distance minimale entre 2 mots de codes

Pour détecter E erreurs, le code complet doit avoir une distance de Hamming $D_h = E + 1$

Pour corriger F erreurs, le code complet doit avoir une distance de Hamming $D_h = 2.F + 1$

3 - Détection des erreurs

Code de contrôle de parité

Contrôle de parité paire ; on ajoute **un bit de parité** au bloc de données

- Si le nombre de bits à 1 dans le bloc de données est pair, le code de parité est 0
- Si le nombre de bits à 1 dans le bloc de données est impair, le code de parité est 1
- Le mot de code (données + bit de contrôle) a donc toujours un **nombre de bits à 1 pair**
- Ex.: données **1010001** => mot de code **10100011**

La distance de Hamming du code de contrôle de parité est 2 :

- Toute erreur simple est détectée.

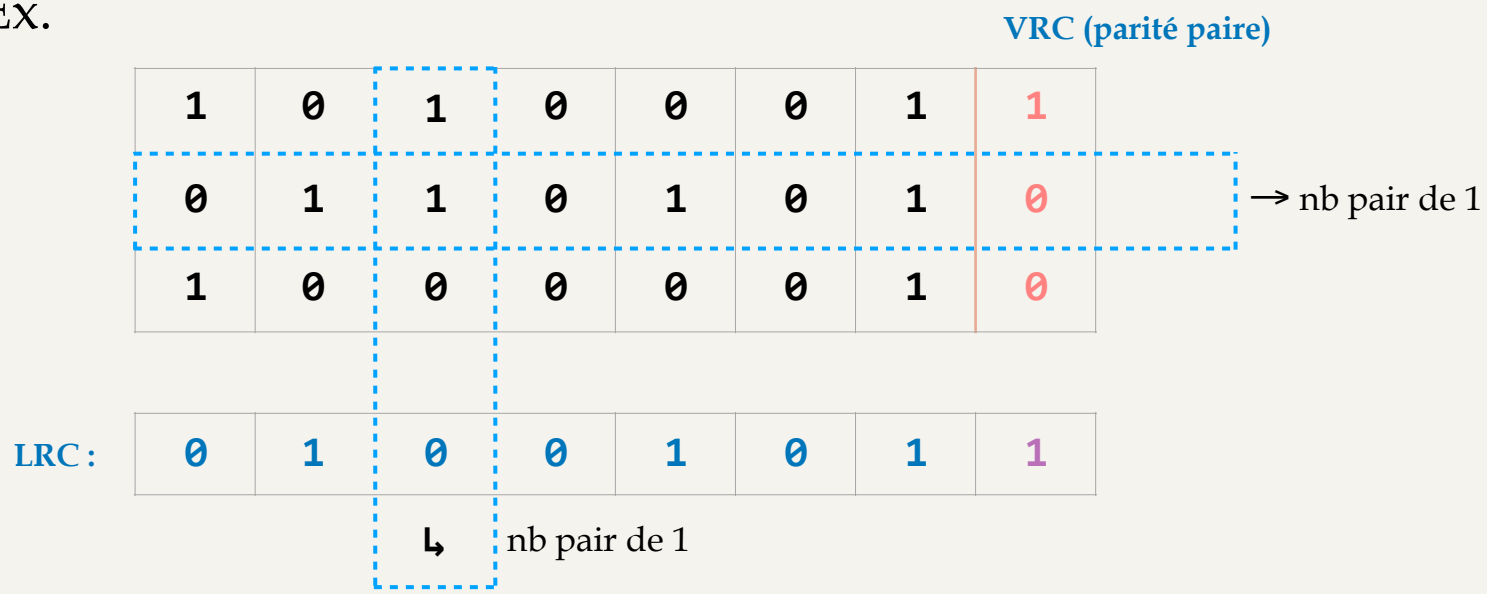
3 - Détection des erreurs

Code de contrôle de parité

On distingue

- des codes de parité verticale, VRC (*Vertical Redundancy Check*),
- des codes parité horizontale, LRC (*Longitudinal Redundancy Check*),
- des codes de parité croisée. Ces derniers permettent la corrections d'erreurs simples.

- Ex.



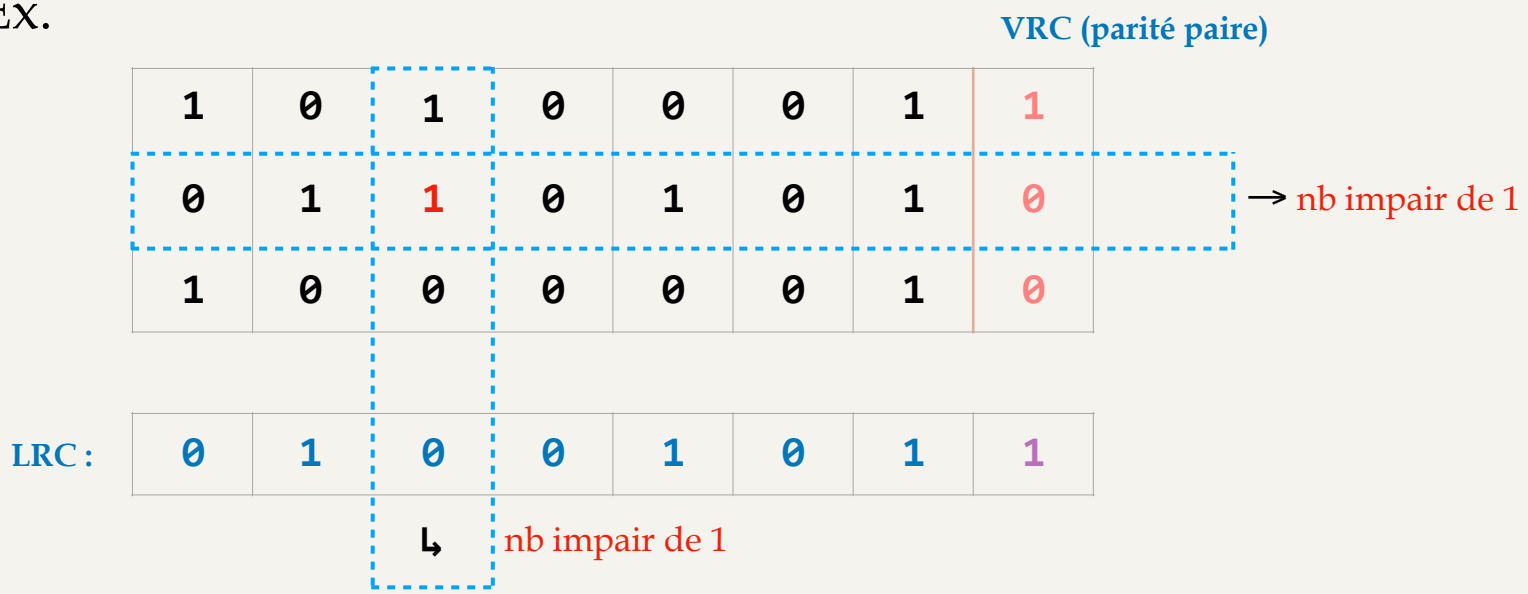
3 - Détection des erreurs

Code de contrôle de parité

On distingue

- des codes de parité verticale, VRC (*Vertical Redundancy Check*),
- des codes parité horizontale, LRC (*Longitudinal Redundancy Check*),
- des codes de parité croisée. Ces derniers permettent la corrections d'erreurs simples.

- Ex.



3 - Détection des erreurs

CRC : Code de redondance cyclique, ou code polynomial

C'est un code classique de **détection** d'erreur.

- On applique une **arithmétique polynomiale modulo 2**
  - L'addition comme la soustraction reviennent à un OU exclusif entre les opérandes
  - La division est réalisée via des soustractions modulo 2
  - Les calculs peuvent aussi se faire à l'aide de polynômes
- Émetteur et récepteur utilisent le même **code générateur G** de g bits
- **M** est le mot de code à transmettre vers le récepteur ; l'émetteur va concaténer à **M** un mot de contrôle **R** de r bits, avec r = g - 1. Le résultat **M•R = T** sera transmis au récepteur qui considèrera **M** comme correct si le reste de la division de **T** par **G** est nul.
  - **G = 10011**
  - **M = 1101011011**
    R = 1110
  - **T = 11010110111110**

Voir l'annexe https://utc505.seancetenante.com/documents/CRC-Code-de-redondance-cyclique.pdf

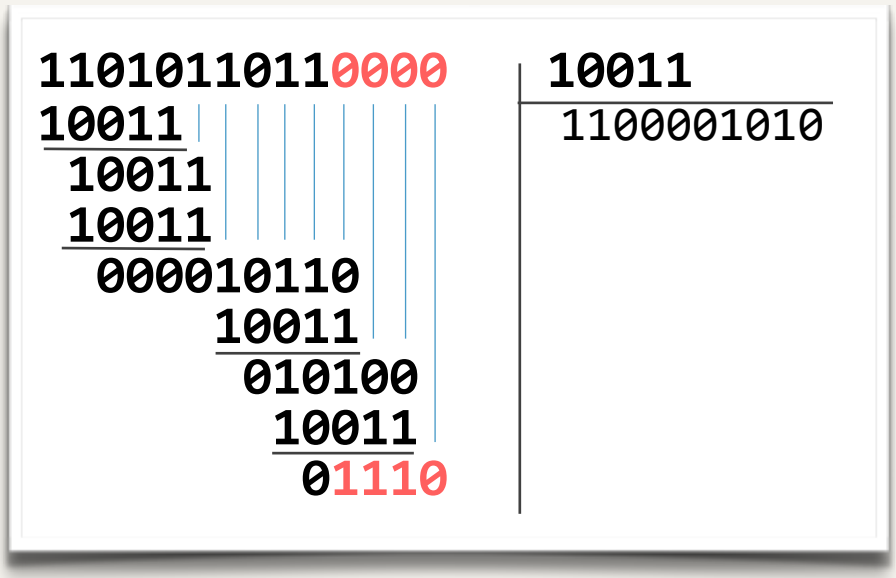


Fig 3.3 - Division pour le calcul du CRC

4 - La sous-couche MAC

Définitions

MAC (*Media Access Control*) est la sous-couche inférieure de la liaison de données

- elle gère l'accès au support physique
- elle règle les problèmes d'adressage (adresses MAC, de 6 octets)
- elle contrôle les erreurs, via un CRC, le FCS (*Frame Check Sequence*)

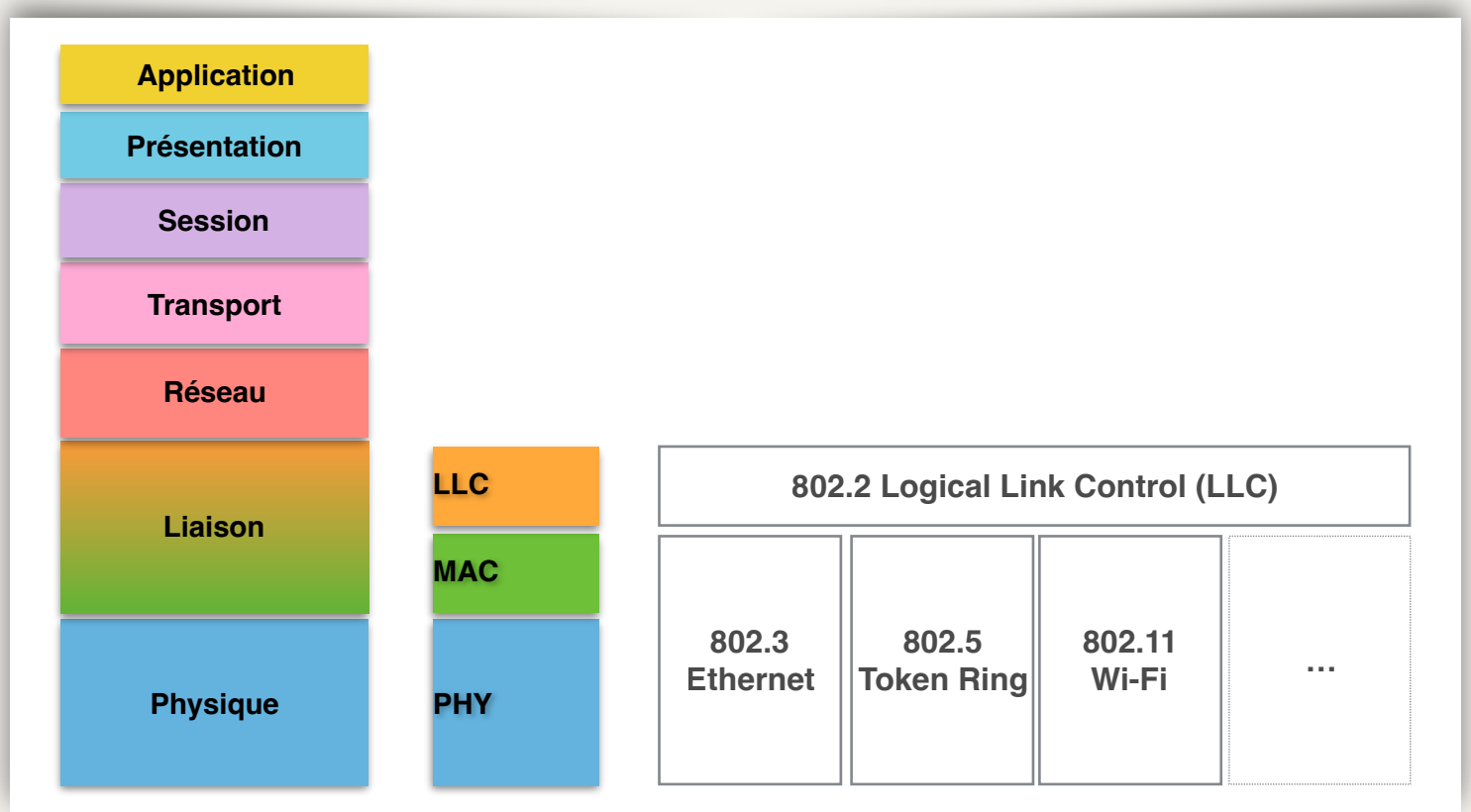


Fig 3.5 - Couches PHY, MAC et LLC

4 - La sous-couche MAC

Définitions

On distingue différentes méthodes d'accès au canal d'un réseau local :

- Les méthodes aléatoires, ou **à contention**
  - **CSMA** (*Carrier Sense Multiple Access*) = Écoute de porteuse avec accès multiple
    - CSMA/CA (*Collision Avoidance*) = **à prévention** de collision
    - CSMA/CD (*Collision Detection*) = **à détection** de collision
- Les méthodes à réservation, ou **à jeton**
  - un jeton (*token*) est une trame qui circule sur le réseau. Une station qui veut émettre doit attendre le jeton, libre, pour le remplacer par sa trame. Lorsque sa trame revient, il la remplace par le jeton.
  - Illustration animée
- Les méthodes à partage de canal, qui exploitent les techniques de multiplexage
  - TDMA (*Time Division Multiple Access*) (Voir Illustration animée)
  - FDMA (*Frequency Division Multiple Access*)
  - CDMA (*Code Division Multiple Access*)

4 - La sous-couche MAC

L'adressage MAC

L'adresse MAC désigne une interface réseau d'un équipement d'une manière unique

Elle est gravée par le fabriquant sur l'adaptateur réseau, NIC (*Network Interface Card*)

Le format universel IEEE nommé MAC-48 ou EUI-48 se compose de 48 bits :

- OUI (*Organization Unit Identifier*) : 3 octets
- SN (*Serial Number*) : 3 octets pour un numéro de série unique donné par le fabriquant
- On exprime en général une adresse MAC avec 6 octets notés en hexadécimal, séparés par « : »

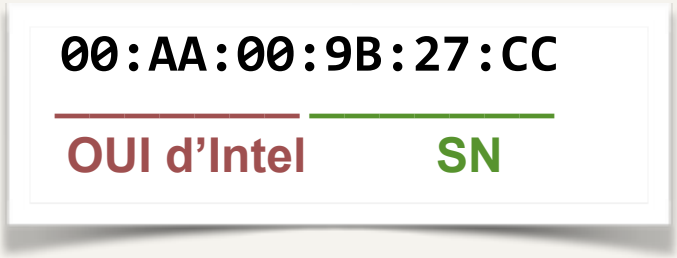


Fig 3.6 - Adresse MAC

- Un broadcast est réalisé avec `FF:FF:FF:FF:FF:FF`
- L'IEEE a défini un autre format d'adresse de 64 bits appelé EUI-64

4 - La sous-couche MAC

CSMA/CD

Carrier Sense Multiple Access / Collision Detection

- ***Carrier Sense*** : écoute de la porteuse (pour vérifier que le support est libre avant d'émettre et pour détecter les colisions)
- ***Multiple Access*** : Accès multiple à un canal unique
- ***Collision Detection*** : détection de collision

Une collision est un **mélange de signaux**

- Elle est détectée par les stations émettrices car elles maintiennent l'écoute du canal pendant l'émission d'une trame
- Quand une station détecte une collision, elle émet une séquence nommée « *JAM signal* », (un signal de brouillage), afin que toutes les stations concernées détectent la collision
- Deux trames victimes de collision devront être retransmises par leur station d'origine

CSMA/CD est utilisé dans des réseaux à diffusion de type Ethernet.

4 - La sous-couche MAC

CSMA/CD

Carrier Sense Multiple Access / Collision Detection

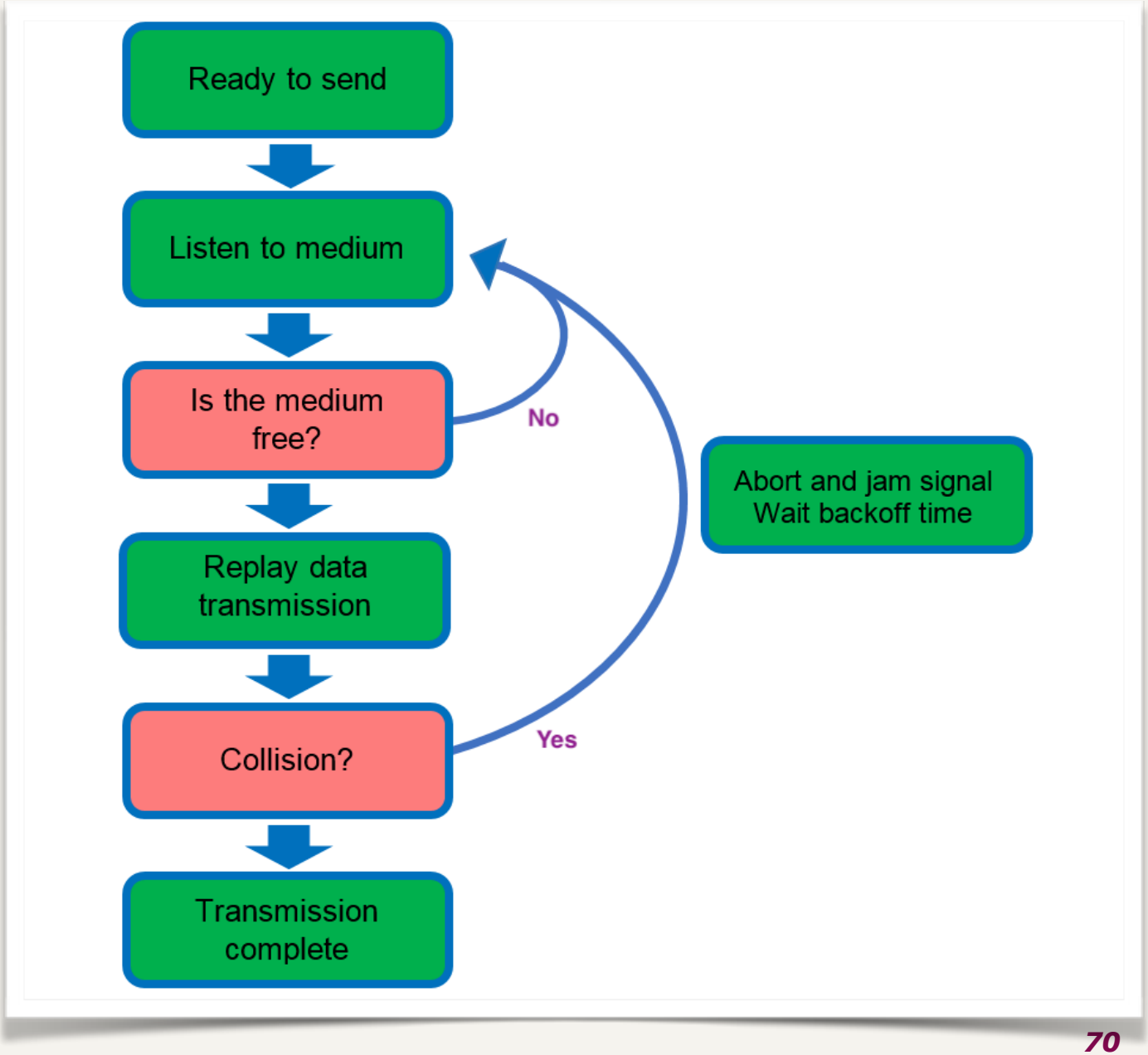


Fig 3.7 - Principe du CSMA/CD

4 - La sous-couche MAC

CSMA/CD

Réseau à diffusion : une trame émise est diffusée à **toutes les stations**

1. Avant d'émettre, la station **écoute le canal**
2. Si le canal est libre :
   - alors commencer l'émission de la trame et **maintenir l'écoute**
   - sinon => *aller en (5)*
3. Pendant l'émission, **si une collision est détectée** :
   Alors :
   - on arrête alors immédiatement l'émission de la trame
   - et on transmet le « *JAM signal* » afin que toutes les stations détectent la collision
   - On attend pendant une durée aléatoire => *aller en (1)*
4. **Sinon** : fin de transmission réussi => avis à la couche supérieure
5. Le canal est occupé => attendre que le canal soit libre
6. Le canal devient libre => attendre une durée aléatoire ; si nombre max. d'essais de transmission non dépassé, *aller en (2)*
7. Le nombre max. d'essais de transmission est dépassé => avis d'échec à la couche supérieure

CSMA/CD

La phase (6) est une phase de **contention**

Des collisions surviennent car deux stations peuvent être en phase (2)

La fenêtre de collision est le temps minimal d'émission pour qu'une collision soit détectée

- Elle vaut deux fois le temps de propagation d'une trame sur la plus grande distance
- Voir l'illustration animée

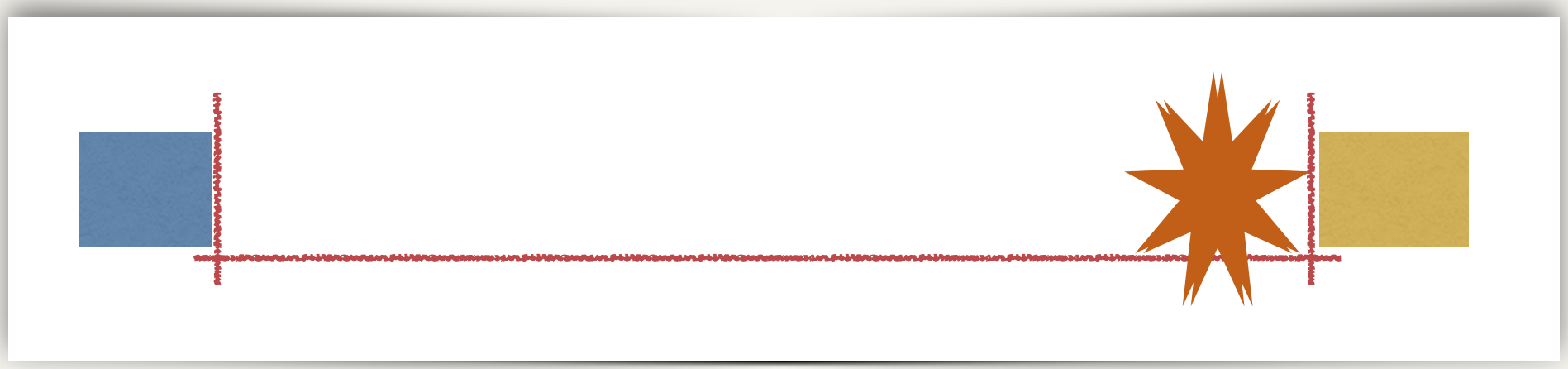


Fig 3.8 - Fenêtre de collision

CSMA/CD

La phase (6) est une phase de **contention**

Des collisions surviennent car deux stations peuvent être en phase (2)

La fenêtre de collision est le temps minimal d'émission pour qu'une collision soit détectée

- Elle vaut deux fois le temps de propagation d'une trame sur la plus grande distance
- Voir l'illustration animée

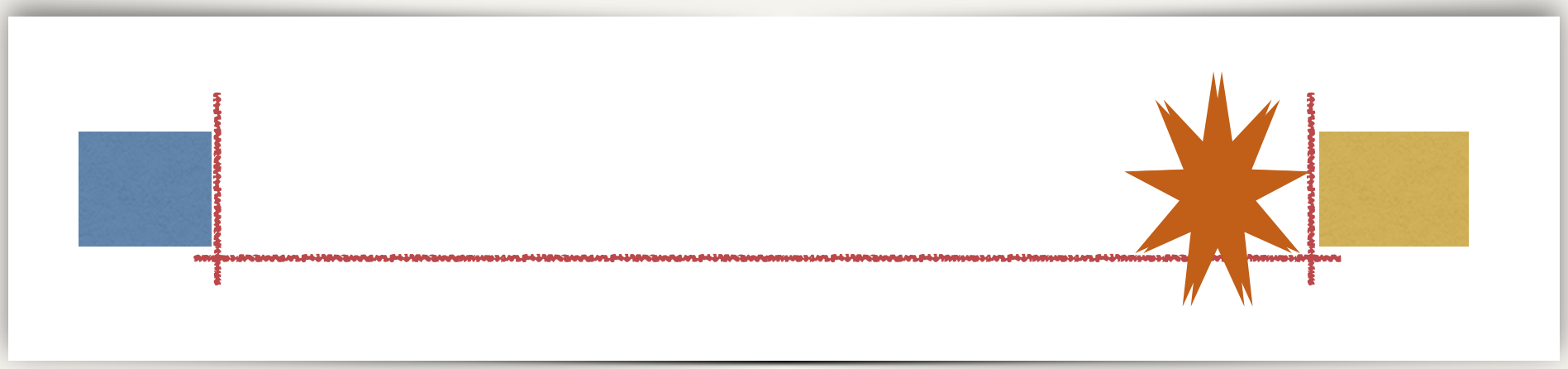


Fig 3.8 - Fenêtre de collision

4 - La sous-couche MAC

Protocole de LAN sans fil

Réseau à diffusion nécessitant des protocoles adaptés

Problème de la **station cachée**

- A et C souhaitent communiquer avec B
- A émet. C est hors de portée de A. Si C émet aussi, des interférences se produisent pour B

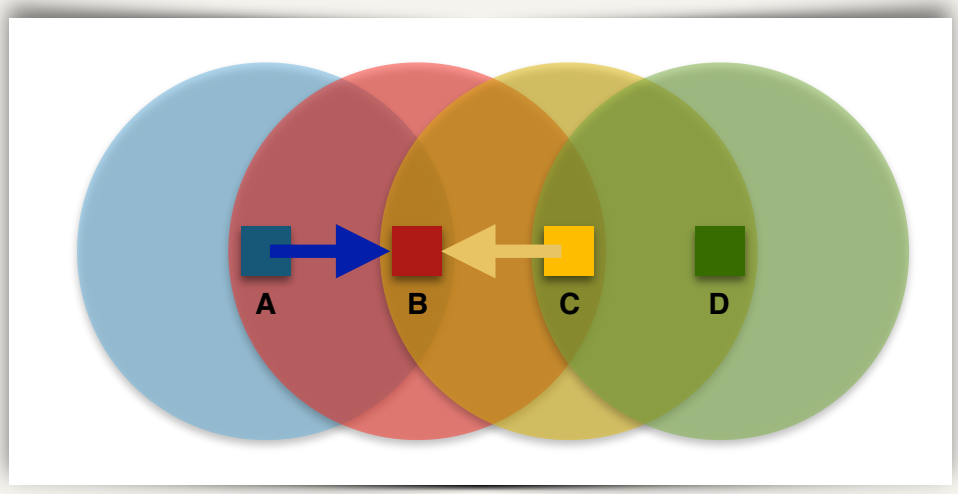


Fig 3.9 - problèmes de la station cachée

4 - La sous-couche MAC

Protocole de LAN sans fil

MACA (*Multiple Access with Collision Avoidance*) est un exemple de protocole

- Lorsque la station B veut envoyer une trame T vers C,
  - elle envoie d'abord une mini trame RTS (*Request to Send*) = « demande d'émission », indiquant la longueur de la trame T qui va suivre
  - C répond à B avec CTS (*Clear to Send*) = « prêt à l'émission », encore avec la longueur de la trame T
  - Quand B reçoit CTS de C, elle commence à émettre T
- Les stations à portée de B reçoivent RTS ; celles qui ne reçoivent pas CTS peuvent émettre à leur tour.
- Les stations à portée de C reçoivent CTS et elles reportent toute transmission éventuelle le temps de la transmission de T de B vers C

IEEE 802.11 (Wi-Fi) utilise une variante de MACA nommée **CSMA/CA** (*Carrier Sense Multiple Access with Collision Avoidance*)

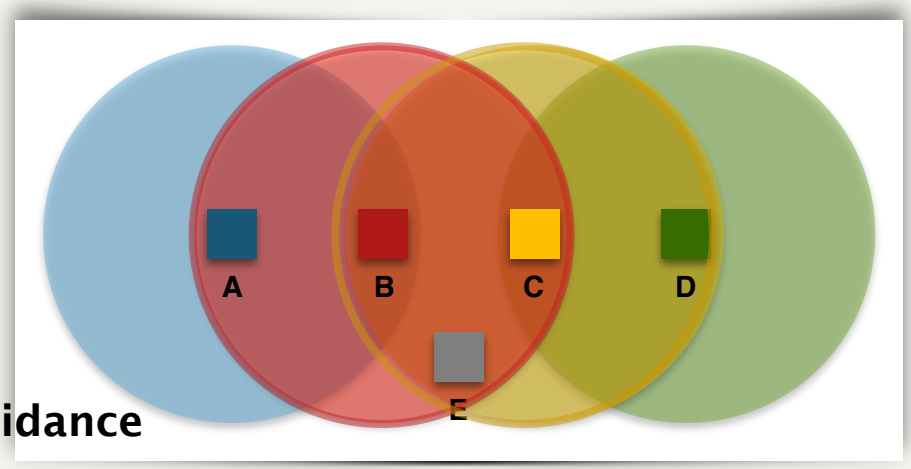


Fig 3.10 - Multiple Access with Collision Avoidance

4 - La sous-couche MAC

Les normes IEEE 802

Les réseaux locaux sont normalisés par l'IEEE (*Institute of Electrical & Electronics Engineers*) via des groupes de travail du comité 802.

Les plus connus sont :

- 802.1 Vue d'ensemble, définitions, architectures des LAN
- 802.2 LLC (Logical Link Control)
- **802.3 Ethernet**
- 802.5 Token-Ring
- **802.11 LAN sans fil (Wi-Fi)**
- 802.15 Réseaux personnels sans fil (Bluetooth)
- 802.16 MAN sans fil (WiMax)

ISO (*International Organization for Standardization*) normalise à son tour certain standards IEEE

4 - La sous-couche MAC

IEEE 802.3 et les réseaux Ethernet

Ethernet commence avec les travaux de **Bob Metcalfe** au Xerox PARC, dans les années 1970

- Normes IEEE 802.3 en 1983
- Années 1990 :
  - Ethernet supplante Token Ring et FDDI
  - Ethernet sur paires torsadées et sur fibre optique
  - Ethernet partagé et **commuté ;** 10BaseT
  - Fast Ethernet (100Base-TX…)
  - Gigabit Ethernet (1000Base-T…)
- Actuellement :
  - **Ethernet commuté** a remplacé l'Ethernet partagé
  - IEEE 802.3an - 10GBase-T et IEEE 802.3ae - 10GBase-F
  - IEEE 802.3bs - 400 Gigabit Ethernet et 200 Gigabit Ethernet (2017)

Fig 3.11 - Bob Metcalfe

4 - La sous-couche MAC

IEEE 802.3 et les réseaux Ethernet

Les premières versions d'Ethernet comme 10Base5 ou 10Base2 utilisaient des **câbles coaxiaux**

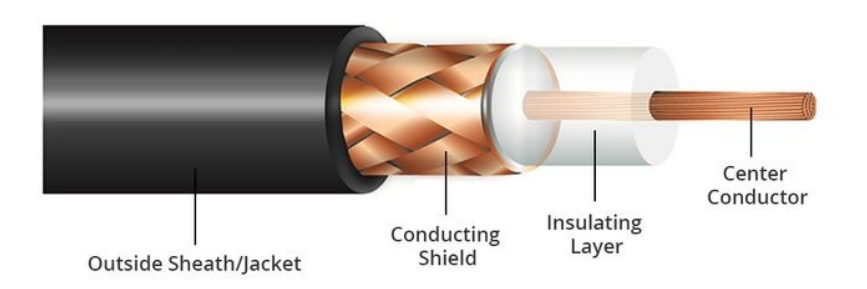


Fig 3.12 - Câble coaxial

Ethernet a beaucoup évolué avec l'utilisation de la **paire torsadée**

Chaque évolution reste compatible avec les normes précédentes et avec la **méthode d'accès CSMA/CD**

Fig 3.13 - Câble à paires torsadées

- **10BaseT**
  - 10 Mbit/s, codage en bande de base (Manchester)
  - T => **Twisted pair** ; une paire pour l 'émission, une pour la réception
  - Topologie en étoile autour d'un **concentrateur** (*hub*)
  - Liaison point à point entre station et hub ; 100 m maximum
  - Connecteur **RJ45**
  - 3 niveaux de hubs maximum

4 - La sous-couche MAC

IEEE 802.3 et les réseaux Ethernet

- **Fast Ethernet** à 100 Mbit/s
  - Norme 802.3u adopté en 1995 ; variantes 100Base-T4, **100Base-TX**, 100Base-FX
  - Remplacement progressif des concentrateurs (*hub*) par des **commutateurs** (*switch*) => un seul domaine de collision par port de switch

- **Gigabit Ethernet**
  - Ratifiée par IEEE, comité 802.3ab, en 1999
  - 1000Base-T utilise en full duplex 4 paires torsadées d'un câble FTP de catégorie 5e ou supérieure, 100 m maximum
    - Compatible avec 100Base-TX et 10Base-T
  - 1000Base-SX : 1 Gbit/s sur fibre optique multimodes à 850 nm
  - 1000Base-LX : fibre optique monomodes et multimodes à 1300 nm

4 - La sous-couche MAC

IEEE 802.3 et les réseaux Ethernet

- **10 Gigabit Ethernet**
  - Normes en 2002 et 2004 pour la **fibre optique** et câbles blindés en cuivre
  - Normes IEEE 802.3ae ou 10GBase-F
    - 7 variantes publiées depuis 2002
    - Fibres multimode et monomode ; Longueurs d'onde 850, 1310 et 1550 nm ; réseaux LAN, MAN, WAN
  - Normes IEEE 802.3an ratifiées en 2006 pour le câble à paires torsadées
    - Câble catégorie 6e, 6a ou 7, en full duplex sur 4 paires
    - 100 m maximum

- **IEEE 802.3ba**
  - Travaux depuis 2007 sur Ethernet à **40 et 100 Gbit/s**
  - Approuvés par l'IEEE en juin 2010
  - Débits jusqu'à 100 Gbit/s :
    - fibre optique monomode, max. 40 km
    - fibre optique multimode, max. 150 m avec OM4, 100 m avec OM3
    - paires torsadées : max. 7 m

4 - La sous-couche MAC

Le protocole MAC IEEE 802.3

Le format de la trame Ethernet (exemple de la trame Ethernet V2)

- Adresses MAC destination (unicast, multicast ou broadcast)
- Adresses MAC source
- Type : pour indiquer au récepteur le protocole lié aux données.
  Ex. 0x0600 : Xerox Network Systems ; 0x0800 : IP ; 0x8100 : 802.1q (encapsulation vlan) ; 0x0806 : ARP (Address Resolution Protocol)
- Pad : 0 à 46 octets de bourrage à 0 afin que la trame fasse au min. 64 octets
- FCS : Frame Check Sequence de type CRC
  $G(x) = x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

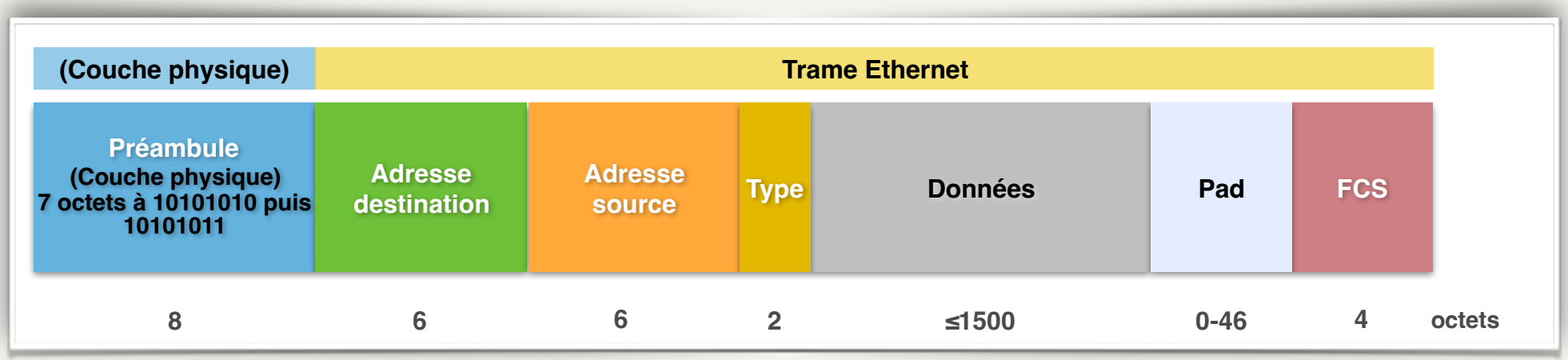


Fig 3.14 - Trame Ethernet V2

Wi-Fi et les normes IEEE 802.11

Les WLAN (*Wireless LAN*) sont très répandus

Ils coexistent avec d'autres technologies sans fil :

- BlueTooth (Dent bleu, celle d'un roi Danois, Harald 1er)
  - Développé en 1994 par Ericsson puis standardisé IEEE 802.15.1
  - Utilise une bande de fréquence autour de 2,4 GHz
  - 1 Mbit/s, portée 20 m env.
  - WPAN ; connectique sans fil

- Réseaux cellulaires 3G, 4G et 5G
  - UMTS, *Universal Mobile Telecommunications System*
  - LTE, *Long Term Evolution*

4 - La sous-couche MAC

Wi-Fi et les normes IEEE 802.11

Les normes IEEE 802.11 - couche PHY

- 802.11b (en 1999) et 802.11g (2003)
- **802.11n** ratifié en 2009
  - Bande à 2,4 GHz ou 5 GHz
  - 200 à 450 Mbit/s
  - portée 50 à 125 m
  - Technologie MIMO (*Multiple-Input Multiple-Output*) qui exploite plusieurs antennes
  - regroupement de canaux radio
- **802.11ac** ratifié en 2014
  - Bande à 5 GHz ; env. 20 canaux de 20 Mhz
  - 433 à 1300 Mbit/s (avec 4 canaux agrégés)
  - portée 30 à 125 m
  - MIMO (Multiple-Input Multiple-Output) (jusqu'à 8 antennes)
  - regroupement de canaux radio
- **802.11ax** ratifié en 2021
  - Bande de 1 à 7,1 GHz ; env. 20 canaux de 20 Mhz
  - 1 à 10 Gbit/s (avec 8 canaux agrégés) ; portée 30 à 125 m
  - MIMO et regroupement de canaux radio : comme 802.11ac

4 - La sous-couche MAC

Wi-Fi et les normes IEEE 802.11

Les normes IEEE 802.11 - couche MAC :

- 802.11e - Ajoute des mécanismes de **QoS** dans les réseaux 802.11
- 802.11i - WPA2 (*Wi-Fi Protected Access*). Mécanismes d'identification et de chiffrement des données (WPA), afin de remplacer l'algorithme initial WEP de la norme 802.11 qui est obsolète
- 802.11h - Conformité aux règlementations européennes

La protection d'un réseau Wi-Fi :

- Un réseau Wi-Fi n'est jamais sûr.
- Les clés WPA et WPA2 ne sont pas inviolables
- En 2018, **WPA3** est standardisé par le consortium Wi-Fi Alliance
- Les box et bornes d'accès Wi-Fi permettent de configurer et surveiller le réseau. On peut ainsi :
  - Savoir qui est connecté ? Adresses IP et MAC des machines connectées
  - Utiliser une clé WPA forte
  - Filtrer les adresses MAC
  - Désactiver DHCP ;
  - Ne pas diffuser le **nom du réseau**, alias **SSID** (*Service Set Identifier*)

5 - Pont et commutateur

Pont

Un pont (*bridge*) est une passerelle agissant en couche liaison de données

Il peut permettre d'interconnecter des LAN
- distants
- de protocoles MAC distincts (si l'adaptation peut se faire au niveau 2)

Les principaux protocoles de pont sont :
- *Spanning Tree Protocol* (algorithme de l'arbre recouvrant) permettant de déterminer une topologie réseau sans boucle (appelée arbre) dans les LAN avec ponts. Il est défini dans la norme IEEE 802.1D
- *Shortest Path Bridging*, spécifié par la norme IEEE 802.1aq, est une technologie pour simplifier la création et la configuration des réseaux, tout en permettant un routage à trajets multiple.

5 - Pont et commutateur

Commutateur Ethernet

Un commutateur Ethernet (*Ethernet switch*) est un **pont multiport**, agissant au niveau 2 du modèle OSI

Le commutateur analyse les trames arrivant sur ses ports d'entrée et filtre les données afin de les aiguiller uniquement sur les ports adéquats (on parle de commutation ou de réseaux commutés)

La commutation est réalisée suivant deux techniques :

- ***Store & Forward*** : la trame est stockée, vérifiée puis retransmise sur un port de sortie
- ***Cut Through*** ou ***Fast Forward*** : le commutateur commence l'envoi de la trame sur un port de sortie dès la lecture de l'adresse destination de la trame

La table de commutation est construite par **apprentissage**

- Lorsqu'une trame est reçue sur un port P, le commutateur examine l'adresse source et met à jour l'entrée (adresse S ; port P) de la table de commutation
- Si la destination est inconnue, on opère par inondation : la trame est transmise vers tous les ports (sauf P)
- Si la destination est connue, la trame est commutée vers le bon port

5 - Pont et commutateur

VLAN ; Virtuel Local Area Network

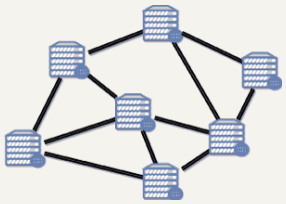
Un réseau local virtuel, (VLAN, *Virtuel Local Area Network*) segmente un réseau physique en plusieurs sous-réseaux logiques, permettant de regrouper des appareils indépendamment de leur emplacement physique.

Fonctionnement :

- Les VLAN sont configurés sur des **commutateurs**, où les ports sont assignés à différents VLAN.
- La communication entre appareils d'un même VLAN est directe, tandis que le trafic entre VLAN nécessite un routeur.

Avantages :

- **Sécurité** : Limite l'accès aux données sensibles à des groupes spécifiques.
- **Performance** : Réduit le trafic de diffusion et optimise l'utilisation du réseau.
- **Flexibilité** : Facilite la gestion et l'organisation des ressources réseau.



Contenu du chapitre

- *Croisements et Destination*
  - Adressage, tables de routage et l'expédition de données dans le réseau IP. Évolution de IPv4 à IPv6.

1 - Objectif de la couche réseau

Objectifs

Couche réseau - Niveau paquet - *Network layer*

- **Acheminer** des paquets de la source jusqu'à la destination
- **Choisir les chemins** appropriés à travers le sous-réseau. Ces chemins passent par des **routeurs**
- Permettre le passage de paquets d'un réseau à un autre
- Gestion du sous-réseau de transport

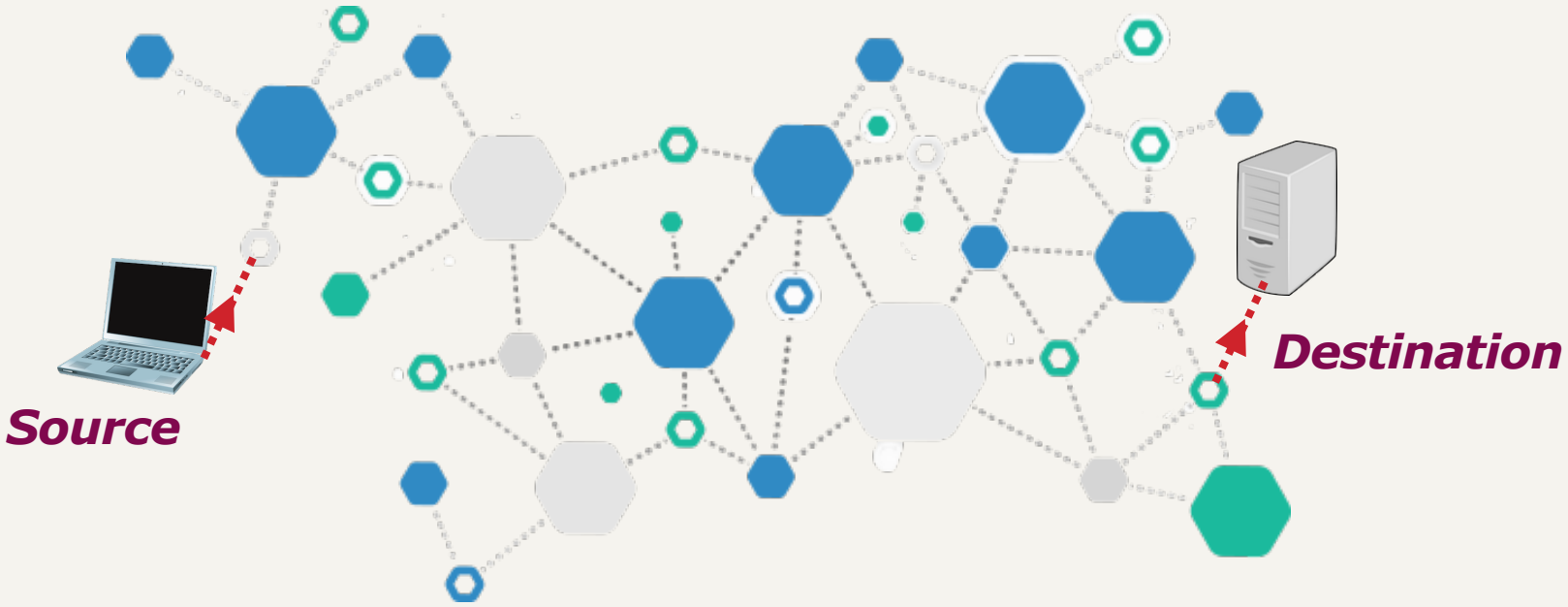


Fig 4.1 - Réseau maillé

1 - Objectif de la couche réseau

Services de la couche réseau

Ils sont fournis à la couche transport

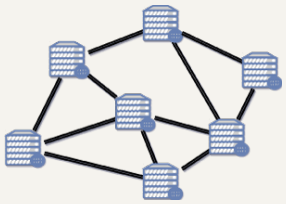
Couche transport	***Client (utilisateur)***
----------	***----------***
Couche réseau	***Opérateur réseau (transporteur)***

Fig 4.2 - Service de la couche réseau

Types de service :

	Mode	
	non connecté	connecté
non fiable	***Mode datagramme*** **1**	~~2~~
fiable	~~3~~	***Circuit virtuel*** **4**

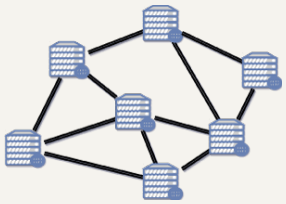
Fig 4.3 - Types de service



1 - Objectif de la couche réseau

Services de la couche réseau

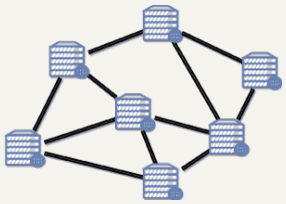
Service non fiable en mode sans connexion		Service fiable orienté connexion
Mode datagramme		**Circuit virtuel**
Service postal	*Analogie*	Service téléphonique
En couche transport (sur les machines d'extrémité)	*Complexité*	en couche réseau (au sein du sous-réseau)
Couche Internet	*Exemple*	ATM ; Relai de trame ; MPLS
Aucune route n'est choisie à l'avance	*Organisation*	La route est choisie à la connexion et mémorisé. Elle est utilisée pour tout le trafic lié à cette connexion



1 - Objectif de la couche réseau

Services de la couche réseau

Service non fiable en mode sans connexion		Service fiable orienté connexion
Mode datagramme		**Circuit virtuel**
Chaque datagramme contient l'adresse de destination et est acheminé indépendamment des autres	*Organisation*	Le CV disparait à la libération de la connexion
Chaque routeur maintient une table de routage (@destination ; ligne de sortie)	*Routage*	Chaque routeur maintient une table de routage (n° CV ; ligne de sortie)
Adaptatif aux défaillances et congestions	*Tolérance aux pannes*	La défaillance d'une route => celle du CV entier
Temps d'analyse des paquets important	*Efficacité*	Analyse des paquets très rapide



2 - Algorithme de routage

Définitions ; introduction

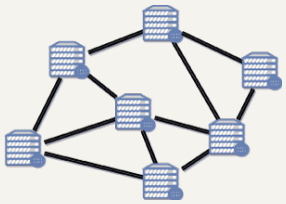
Chemin : suite de liens et de nœuds intermédiaires parcourus pour aller de la source à la destination dans le réseau

Un **algorithme de routage** est la partie du logiciel de réseau responsable du **choix d'une ligne de sortie** d'un routeur en fonction de la destination d'un paquet entrant

À cette fin, chaque routeur gère **une table de routage**

On distingue :

- Des algorithmes non adaptatifs
  - Le routage est **statique**
  - Les routes sont calculées à l'avance
  - Cf. commande **route** des systèmes Unix et Linux
- Des algorithmes adaptatifs
  - Le routage est **dynamique**
  - Les décisions de routage sont modifiées en fonction de changements (trafic, topologie, etc.)



Définitions ; introduction

La **métrique** utilisée est une fonction de :

- La distance géographique
- Le nombre de sauts
- Le temps d'acheminement (temps de transit + délais d'attente dans les routeurs)
- Le coût de transport
- …
- Ou bien une **fonction pondérée** de variables ci-dessus

La **table de routage** est utilisée pour la fonction de **relayage** de paquets

- Pour **acheminer** le paquet vers le prochain saut
- FIB, *Forwarding Information Base*, table d'information d'acheminement

RIB, *Routing Information Base*, table de routage :

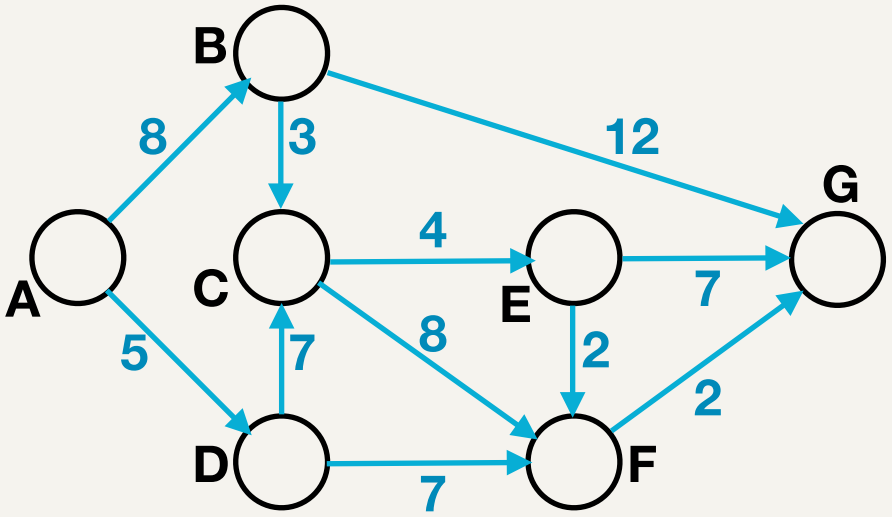
- Permet de **calculer** et de choisir les chemins
- RIB est mis à jour par l'algorithme de routage

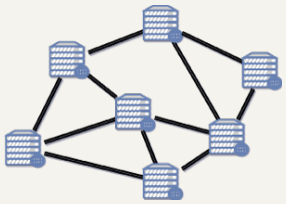
Voir : https://www.youtube.com/watch?v=EPo7QyB7Yss

Exemples de routage

Routage du plus court chemin - *Shortest Path Routing*

- Algorithme de **Dijkstra**
- Voir https://licence-math.univ-lyon1.fr/lib/exe/fetch.php?media=gla:dijkstra.pdf
- Exercice :
- Voir https://utc505.seancetenante.com/documents/Exercices-UTC505-algorithme-de-Dijkstra.docx
  - Calculer le plus court chemin entre A et G.
    - Par quels nœuds passe ce chemin ? Quel coût a ce chemin entre A et G ?

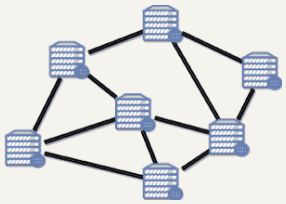




Exemples de routage

Routage à vecteur de distance - *Distance Vector Routing*

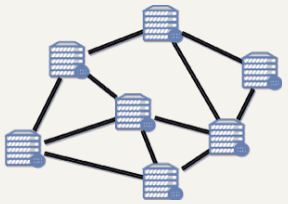
- Ce routage dynamique a été utilisé dans **Arpanet**
- Il reste utilisé avec **RIP** (*Routing Information Protocol*)
- Un vecteur de distance est, pour un routeur R et une destination N connue :
  - $V_{RN} = [\ d_{RN}\ ,\ L_{RN}\ ]$
    avec d_{RN} : meilleure distance connue et L_{RN} : la ligne pour atteindre N
- Chaque routeur R du réseau maintient sa table de routage :
  - $[\ N\ ,\ V_{RN}\ ]$
    soit $[\ N\ ,\ d_{RN}\ ,\ L_{RN}\ ]$
  - et la diffuse aux routeurs voisins
- Chaque nœud R :
  - Apprend ainsi ce que chaque voisin V peut atteindre
  - Met à jour sa propre table :
    - Ajout d'une entrée si le voisin indique une nouvelle destination
    - Calcul et comparaison pour les destinations connues
    - Si $d_{RN} > d_{VN} + d_{RV}$ alors l'entrée $[\ N\ ,\ d_{RN}\ ,\ L_{RN}\ ]$ est remplacé par $[\ N\ ,\ d_{VN} + d_{RV}\ ,\ L_{RV}\ ]$



Exemples de routage

- Voir www.youtube.com/watch?v=kzablGaqUXM

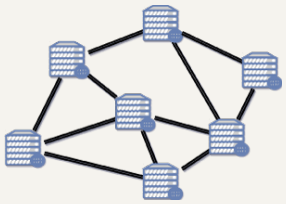
- Ce routage à vecteur de distance doit être amélioré pour assurer une convergence plus rapide et pour éviter la création de boucle dans le réseau.
- On utilise pour cela la technique de l'horizon coupé (*Split Horizon*)



Exemples de routage

Routage par information d'état de liens - *Link State Routing*

- Chaque routeur R doit :
  - Découvrir ses voisins ; un voisin V => un lien R->V
  - Déterminer la distance de chaque voisin : d_{RV}
  - Construire un paquet d'**information d'état de lien** [R , V , d_{RV}]
  - À chaque changement significatif, R ne diffuse que les modifications d'information d'état de liens qu'il a détecté. La diffusion concerne un sous-réseau nommé aire ou zone (*area*)
  - Chaque nœud R entretien une table de routage composée de rangées [D , V , d_{RD}] = Nœud destination, Nœud suivant, coût total
    et la réception de paquet d'information d'état de lien implique la mise à jour de la table suivant l'algorithme de Dijkstra
- **OSPF** (*Open Shortest Path First*) est un routage d'internet qui utilise ce routage par information d'état de liens



Exemple de table de routage

Exemple d'une table de routage IPv4 sur un ordinateur (192.168.0.100) connecté à Internet via une box (192.168.0.1)

Réseau destination (format CIDR)	Masque	Passerelle	Interface	Métrique
0.0.0.0/0	0.0.0.0	192.168.0.1	192.168.0.100	1
127.0.0.0/8	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0/24	255.255.255.0	192.168.0.100	192.168.0.100	1
192.168.0.100/32	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.1/32	255.255.255.255	192.168.0.100	192.168.0.100	1

Exemple de table de routage

Route par défaut

- Réseau destination : **0.0.0.0/0**
- Il s'agit de la route par défaut, utilisée lorsqu'aucune autre route ne correspond
- Tous les paquets qui ne correspondent à aucune autre entrée seront envoyés au routeur 192.168.0.1 via l'interface 192.168.0.100

Réseau de bouclage (Loopback)

- Réseau destination : **127.0.0.0/8**
- Cette entrée gère le trafic pour le réseau de bouclage (localhost)
- Les paquets destinés à ce réseau sont acheminés vers l'interface 127.0.0.1

Réseau local

- Réseau destination : **192.168.0.0/24**
- Cette entrée concerne le réseau local 192.168.0.0/24
- Les paquets destinés à ce réseau sont acheminés directement via l'interface 192.168.0.100

Adresse IP spécifique

- Réseau destination : **192.168.0.100/32**
- Cette entrée est spécifique à l'adresse IP 192.168.0.100
- Les paquets destinés à cette adresse sont acheminés via l'interface de bouclage 127.0.0.1

Exemple de table de routage

Adresse IP spécifique

- Réseau destination : **192.168.0.1/32**
- Cette entrée est spécifique à l'adresse IP 192.168.0.1 (probablement la passerelle par défaut)
- Les paquets destinés à cette adresse sont acheminés via l'interface 192.168.0.100

Informations complémentaires

- Le masque indique quels bits de l'adresse IP correspondent au réseau.
- La passerelle est l'adresse IP du prochain saut pour atteindre le réseau de destination.
- L'interface est l'interface réseau utilisée pour envoyer les paquets.
- La métrique (toujours 1 ici) est utilisée pour déterminer la meilleure route lorsque plusieurs routes sont disponibles.

Protocoles de routage

RIP (*Routing Information Protocol*) ; RFC 1058 et RFC 1721 à 1723 pour RIP-2

- Protocole simple parfois utilisé en Intranet
- Anciennement utilisé dans internet, mais remplacé par les protocoles ci-dessous.

OSPF (*Open Shortest Path First*)

- Protocole de routage interne IP, de type 'à état de lien de liens'.
- OSPFv2 est décrite dans la RFC 2328 en 1997
- OSPFv3 permet l'utilisation d'OSPF dans un réseau IPv6. Voir RFC 2740

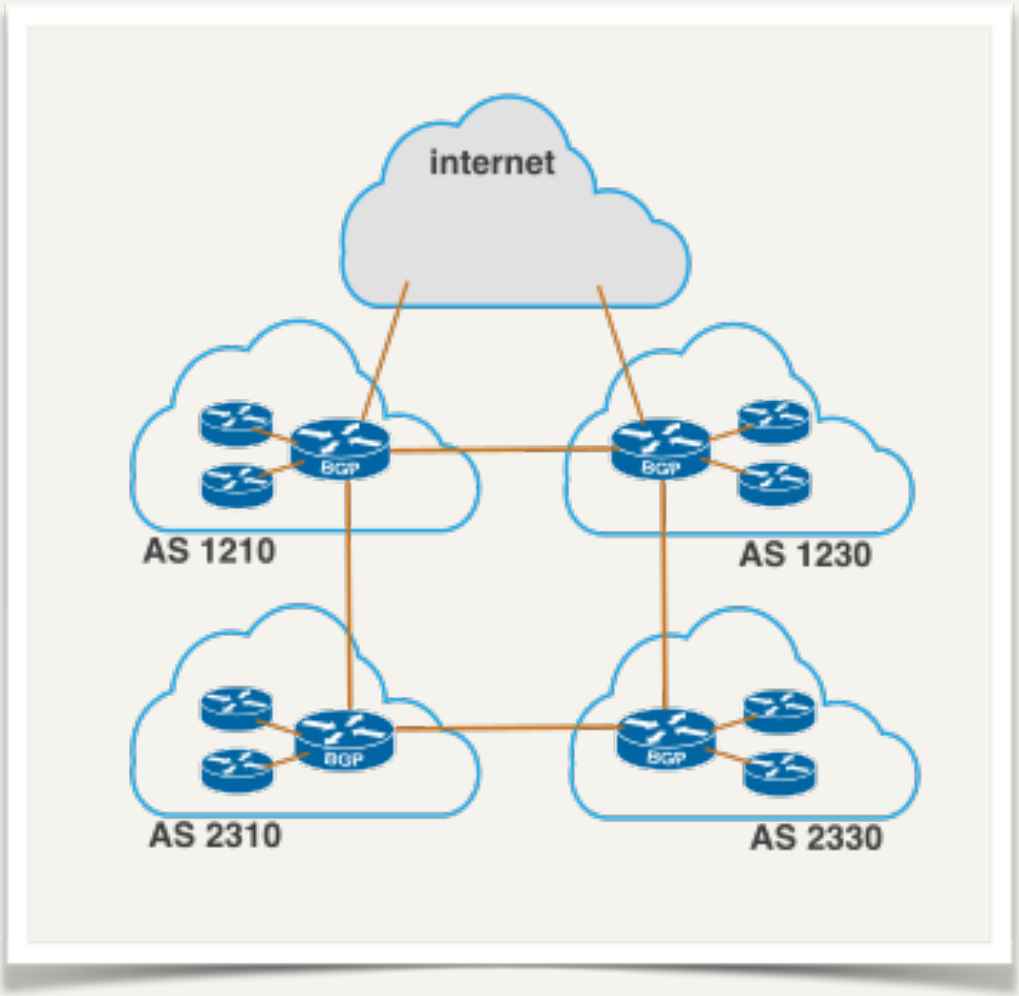
IS-IS (*Intermediate system to intermediate system*)

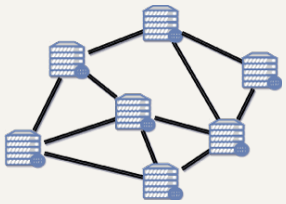
- Protocole de routage interne multi-protocoles à état de liens
- Norme ISO/CEI 10589:2002 également publié par l'IETF avec la RFC 1142
- IS-IS est un protocole à état de liens utilisé à l'intérieur d'un *autonomous system*. Il est apprécié dans des grands réseaux de fournisseurs de services.

Protocoles de routage

BGP (*Border Gateway Protocol*)

- Protocole d'échange d'informations de routage entre *Autonomous Systems* (AS). RFC 4271.
- BGP prend en charge le routage sans classe et utilise l'agrégation de routes afin de limiter la taille de la table de routage.
- La stratégie de routage tient compte de contraintes politiques, économiques ou de sécurité
- BGP est principalement utilisé entre les opérateurs et fournisseurs d'accès à Internet pour l'échange de routes, à travers des services de transit ou de peering.





Besoins d'interconnexion

Lorsque plusieurs réseaux sont interconnectés, on parle d'**inter-réseau** ou de réseau de réseaux

- Le terme **internet** provient justement d'**internetwork**

Grande diversité des réseaux, des technologies et des besoins d'inter-connexion

- Type de service (avec ou sans connexion)
- Protocole (IP, IPX, ATM, MPLS, etc.)
- Adressage
- Taille de paquets
- Qualité de service
- Contrôle de flux et de congestion
- Sécurité (règles de confidentialité, chiffrement, etc.)
- Facturation

Équipements d'interconnexion

Au dessous de la couche réseau, on trouve :

- Les **répéteurs** et les **hubs** (couche physique)
- Les **ponts** et les **commutateurs** (couche liaison de données)
  - Copier et faire suivre des trames

Les passerelles opèrent à des couches supérieures à la couche réseau

Les équipements qui opèrent au niveau de la couche réseau sont des **routeurs**

- Le routeur utilise des adresses logiques (de niveau 3), indépendantes des adresses physiques

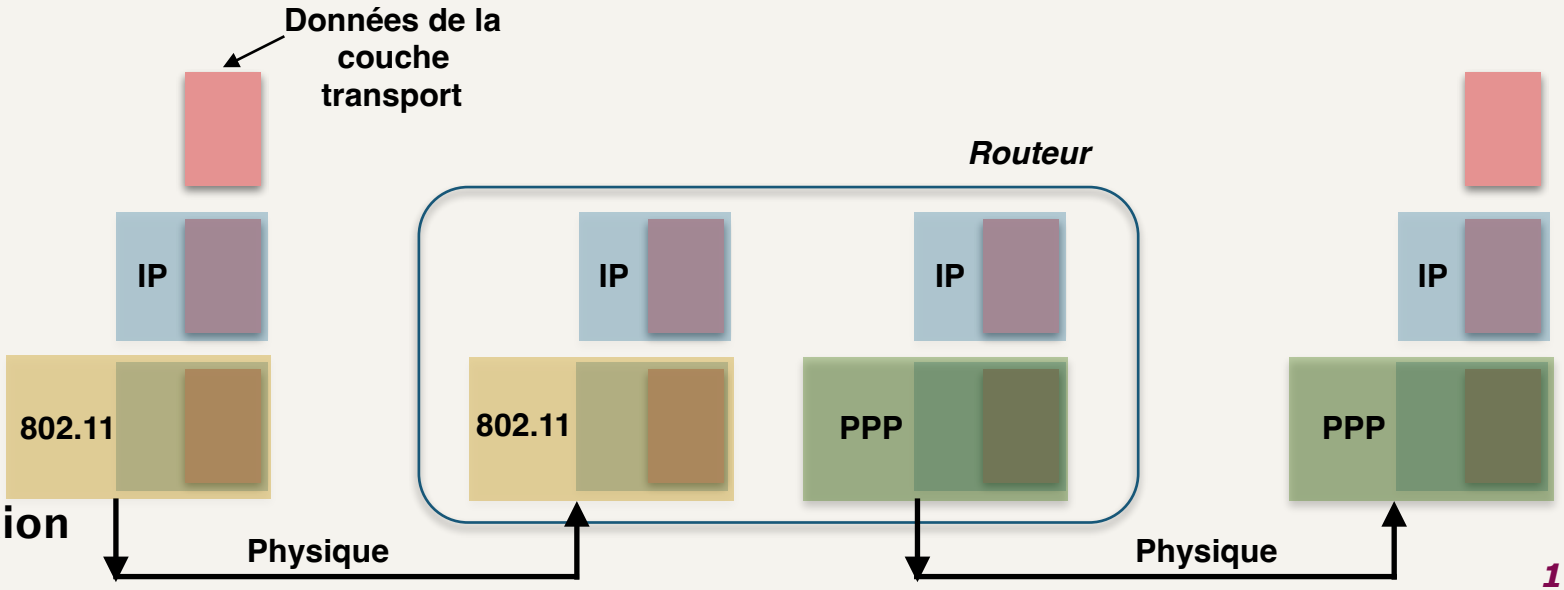
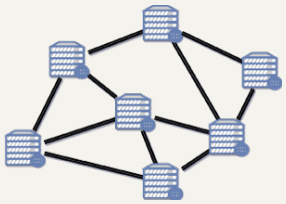


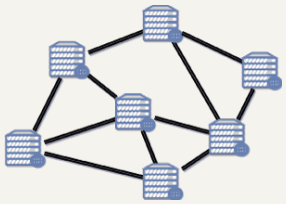
Fig 4.4 - Interconnexion avec un routeur



Équipements d'interconnexion

La **technique du tunnel**

- Les machines d'extrémité sont sur un réseau de même type, mais elles sont séparées par un réseau différent
- Une solution d'interconnexion passe par la technique du tunnel ou le paquet source est encapsulé par le routeur qui ajoute son entête de niveau 3
- Cette technique est souvent utilisée au niveau de couche supérieure (**VPN** = *Virtual Private Network*)



Architecture TCP/IP

- **Dans cette partie, ou presque :**
  - Historique *(recherche personnelle à faire)*
  - La couche Internet ; le protocole IP
  - Les adresses IP v4
  - Sous réseaux
  - IP v6
  - Autres protocoles de la couche internet
  - La couche transport de TCP/IP (Ch. 5)

Couche internet

- Nommée couche ***internet*** ou couche ***inter-réseau***
- Au même niveau de la couche **réseau** d'OSI
- Objectif :
  - Permettre aux hôtes d'introduire des paquets nommés datagrammes sur n'importe quel réseau
  - Acheminer ces datagrammes indépendamment les uns des autres jusqu'à destination.

- Des datagrammes peuvent arriver dans un ordre différent, ou se perdre

Couche internet

- La couche internet définit :
  - Un **protocole** nommé **IP** (*Internet Protocol*)
  - Le format des datagrammes IP
  - Un protocole compagnon, **ICMP** (*Internet Control Message Protocol*), qui assure le routage des datagrammes et la gestion des congestions

- Le protocole IP (*Internet Protocol*) est **non fiable**, **sans connexion**
- Si une fiabilité est nécessaire pour le transfert de données, elle sera assurée par le protocole TCP (*Transmission Control Protocol*) de la couche transport

L'en-tête du datagramme IPv4

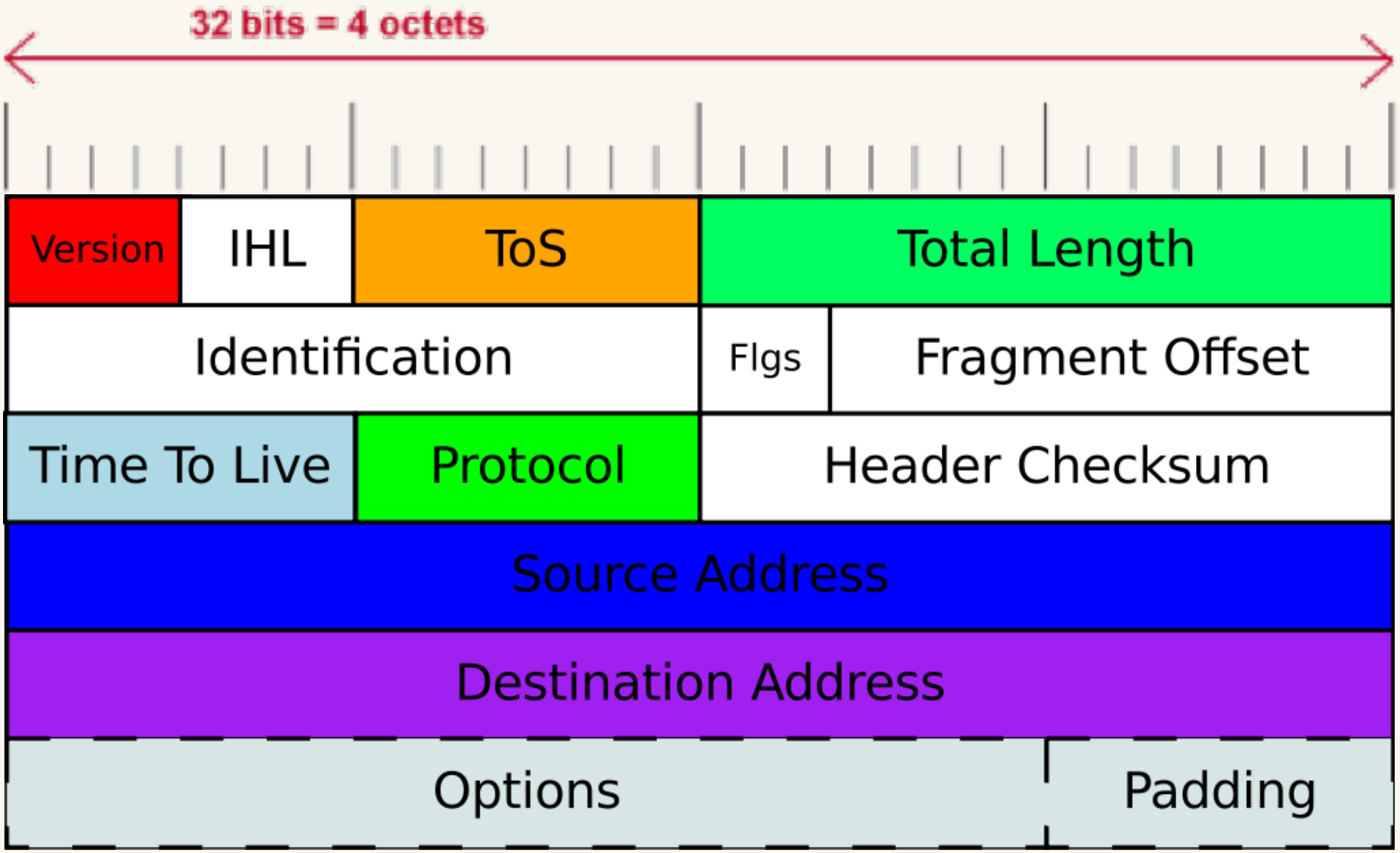


Fig 4.5 - L'en-tête du datagramme IPv4

Le datagramme IPv4

- L'en-tête est de 20 à 60 octets
  - **Version IP (4 bits) = 4**
  - **IHL**, *Internet Header Lenght ; l*ongueur en-tête en nombre de mots de 32 bits (4 bits)
  - **ToS**, Type de service ; (8 bits)
  - **Longueur totale, en octets, du datagramme (16 bits)**
  - Identification ; identifier les fragments d'un paquets(16 bits)
  - *Flgs* : Indicateurs ou Flags (3 bits)
  - *Fragment offset* (13 bits)
  - **TTL, Time To Live : Durée de vie (8 bits)**
  - **Protocol** ; n° de protocole de transport (8 bits)
  - *Header Checksum* ; somme de contrôle de l'en-tête (16 bits)
  - **Adresse source (32 bits)**
  - **Adresse destination (32 bits)**
  - Options (0 à 40 octets)
  - *Padding* ; remplissage ; tel que l'en-tête soit un multiple de 32 bits

L'en-tête du datagramme IPv6

- Un en-tête IPv6 de 40 octets

Adresses IPv6 de 16 octets, soit 128 bits

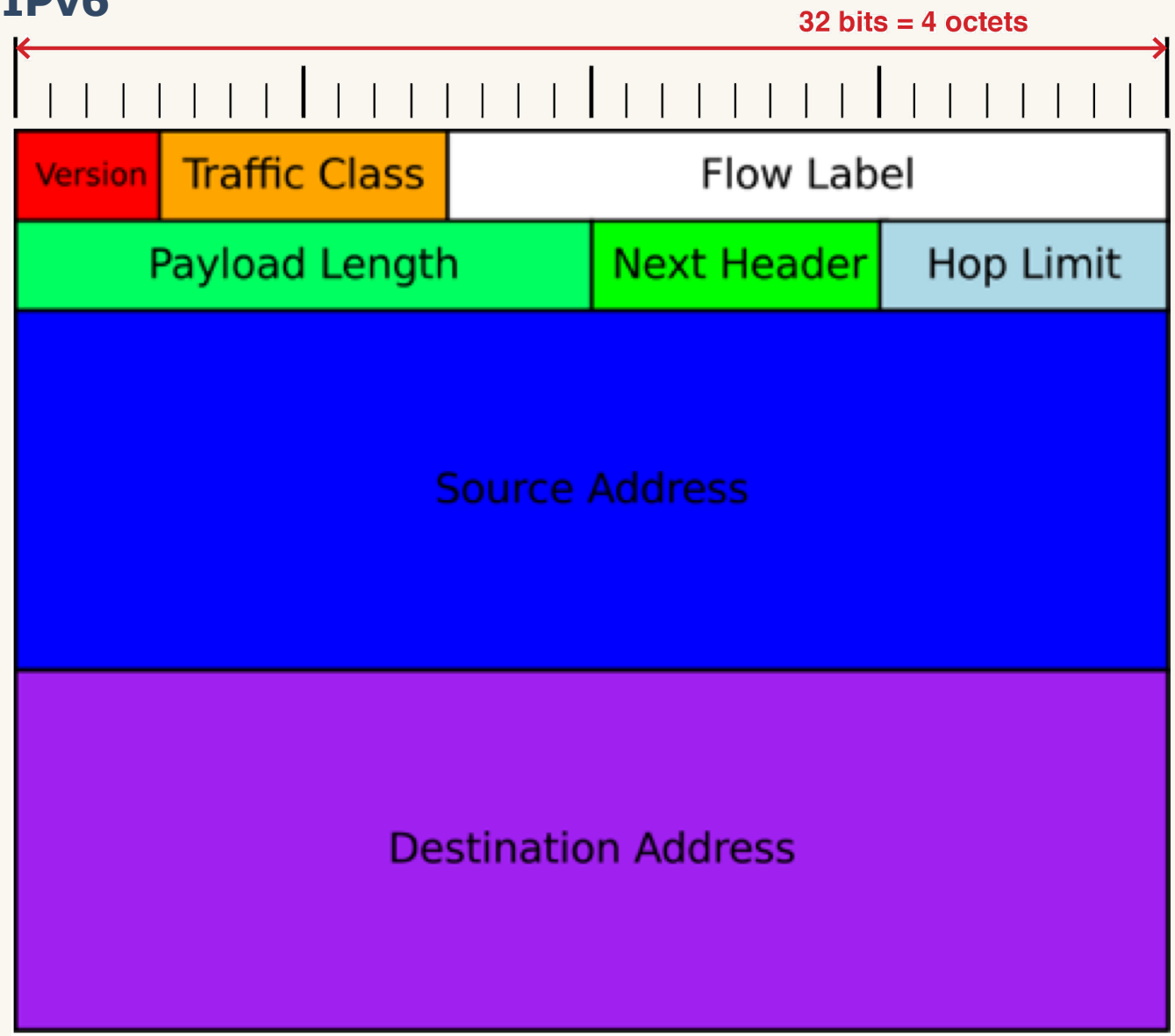


Fig 4.6 - L'en-tête du datagramme IPv6

L'en-tête du datagramme IPv6

- **Version IP** (4 bits) = 6
- *Traffic class* : Classe de trafic ; QoS et indication de congestion (8 bits)
- *Flow label* : Marquage de flux (20 bits)
- *Payload length* : Taille de la charge utile en octets (16 bits)
- *Next header* : Type de l'en-tête suivant (8 bits)
- *Hop limit* : TTL ou Time To Live ou nombre limite de sauts (8 bits)
- **Adresse source** (128 bits)
- **Adresse destination** (128 bits)

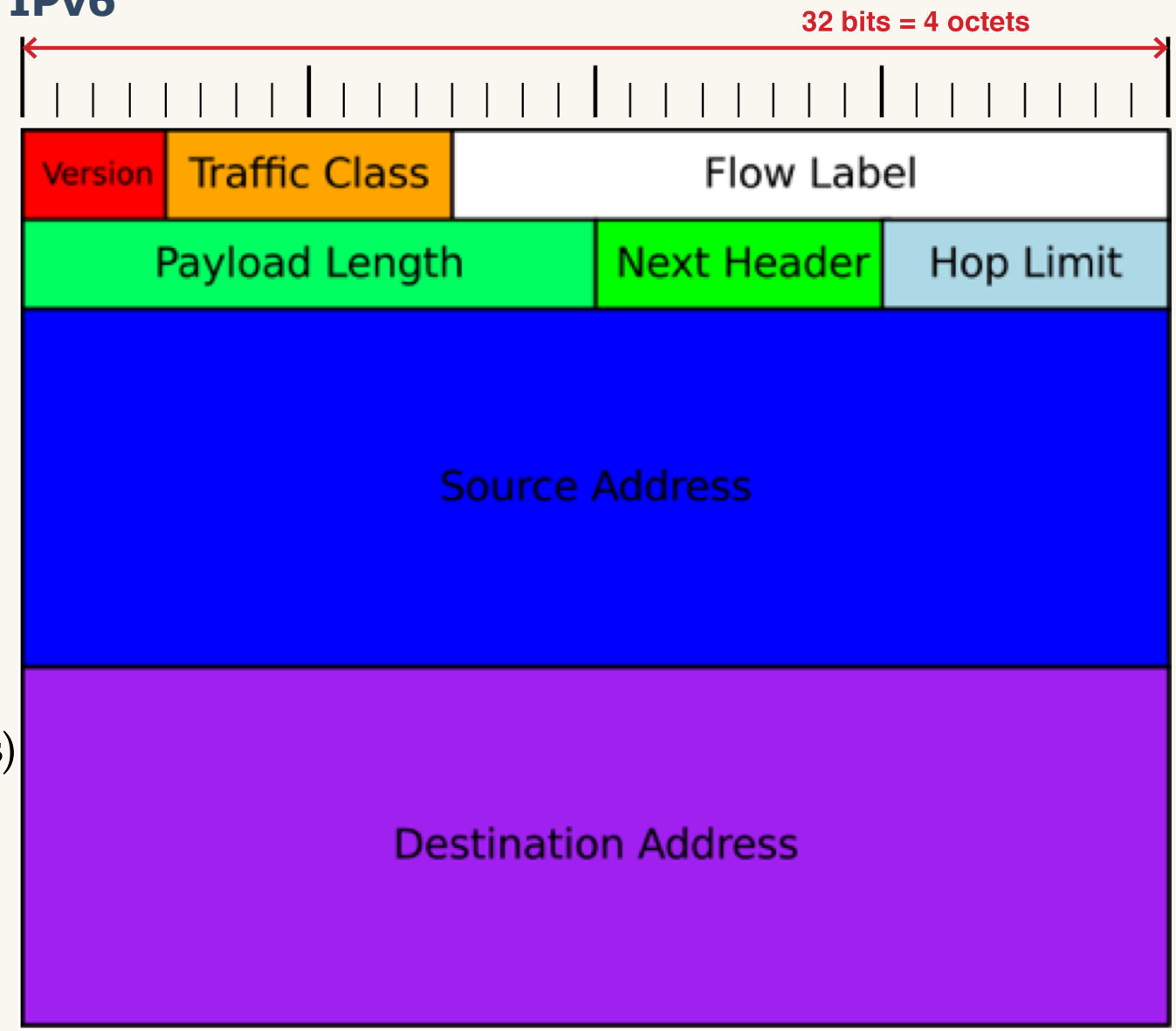
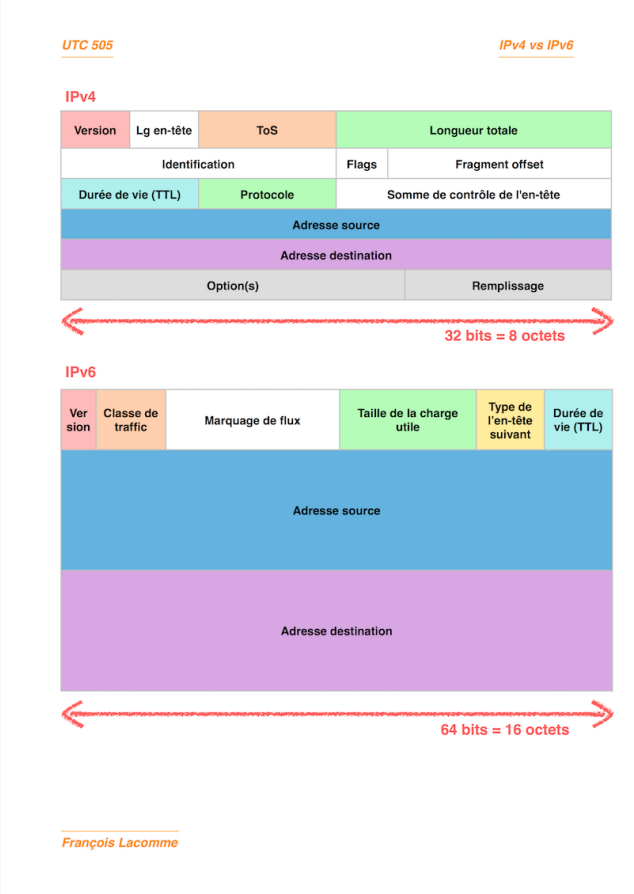


Fig 4.6 - L'en-tête du datagramme IPv6

Comparaison des en-têtes IPv4 et IPv6

- Voir : utc505.seancetenante.com/documents/IPv4-vs-IPv6.pdf



Introduction

- Une adresse IPv4
  - Une valeur sur **32 bits** (4 octets) qui identifie **de façon unique** chaque interface réseau d'**un réseau** TCP/IP
  - Par convention on exprime une adresse IPv4 avec la notation **décimale pointée** :
    4 octets exprimés en décimal, séparés par des points « . »

- Exemple 1 :
  - `192.41.6.120` en décimal pointé
  - `11000000 00101001 00000110 01111000` en binaire

- `41 = 128x0 + 64x0 + 32x1 + 16x0 + 8x1 + 4x0 + 2x0 + 1x1`
- Comment convertir du décimal en binaire :
  - fr.wikihow.com/convertir-du-décimal-en-binaire

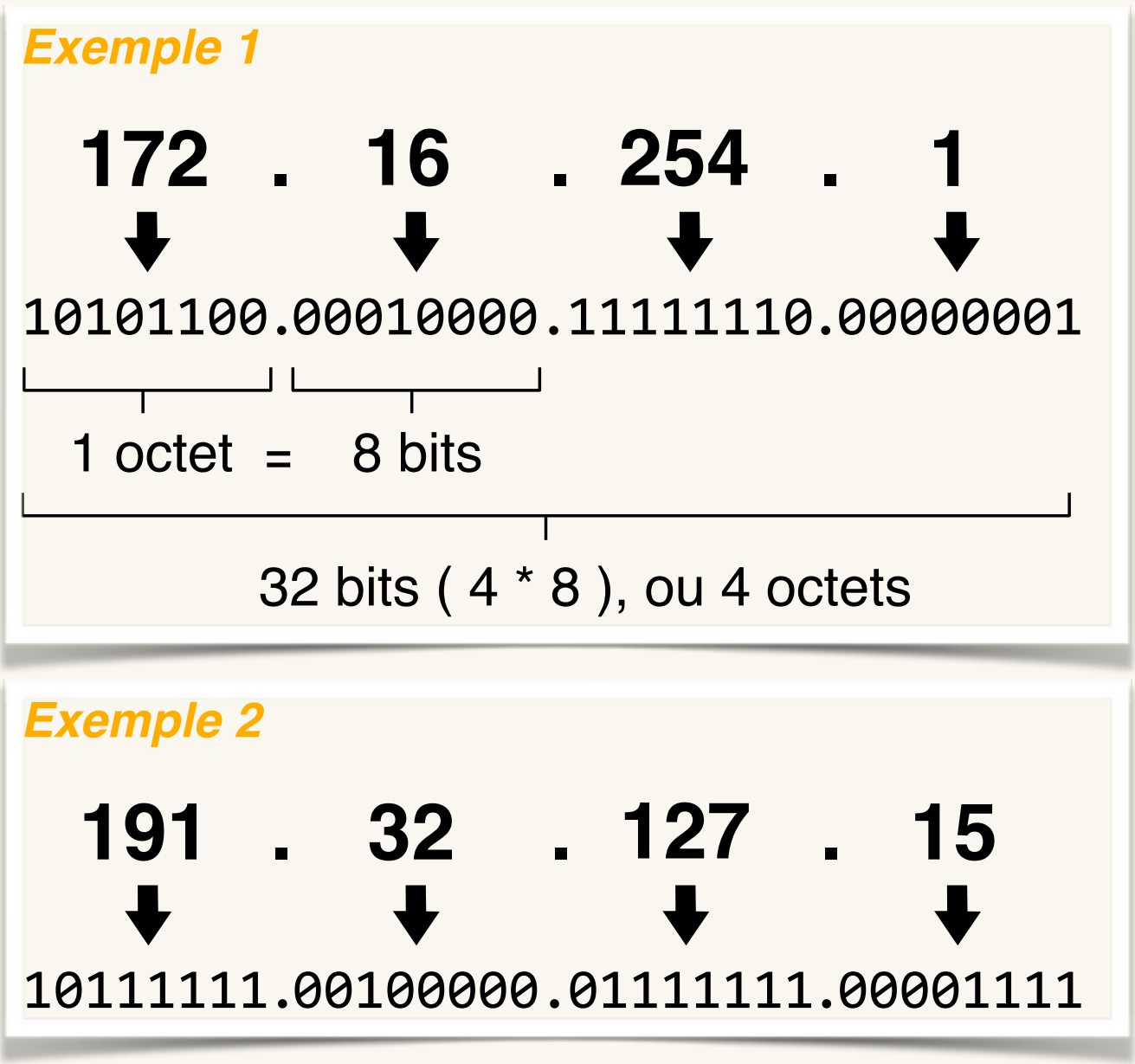


Fig 4.7 -Deux exemples de conversion d'adresse IP (décimal vers binaire)

Structure d'une adresse IPv4

- Une adresse IPv4 permet d'identifier :
  - un **réseau**
  - **l'interface réseau** d'un équipement sur **ce réseau** [on parle, improprement, d'hôte (*host*)]

- Pour déterminer l'**identifiant réseau**, on doit connaitre le **masque d'adresse**
- Un masque d'adresse est **une suite de bits à 1** suivie d'**une suite de bits à 0** (le tout sur 32 bits)

Masque d'adresse

- On exprime un masque d'adresse :
  - soit avec la **longueur du préfixe**, donc le nombre de bits à 1 (ex. /20)
    - Cette longueur du préfixe est souvent abrégé en '**préfixe**'
    - On parle de **notation CIDR** (*Classless Inter-Domain Routing*)
  - soit en **décimal pointé** (ex. 255.255.240.0)

- Un **ET logique** (∧) entre une adresse et un masque permet de déterminer **l'identifiant réseau**

- L'**adresse IPv4 ∧ le complément à un du masque** détermine **l'identifiant d'une interface-réseau** (ou, improprement, l'identifiant d'hôte [*host ID*])

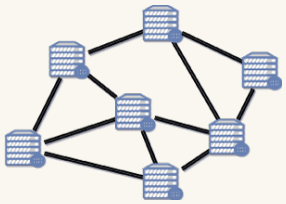
- Exemple 1
  - Identifiant réseau

```
   172. 16.254.   1     10101100.00010000.11111110.00000001
∧  255.255.255.   0     11111111.11111111.11111111.00000000
   ____________________  ___________________________________
=  172. 16.254.   0     10101100.00010000.11111110.00000000
```

  - Identifiant d'hôte *(ou plutôt d'interface réseau)* dans un réseau

```
   172. 16.254.   1     10101100.00010000.11111110.00000001
∧    0.   0.   0.255     00000000.00000000.00000000.11111111
   ____________________  ___________________________________
=    0.   0.   0.   1     00000000.00000000.00000000.00000001
```


Fig 4.8 - Exemple 1 de calcul des identifiants réseau et hôte d'une adresse IPv4



- Exemple 2
  - Identifiant réseau

```
   191. 32.127.15     10111111.00100000.01111111.00001111
∧  255.255.240. 0     11111111.11111111.11110000.00000000
  ___________________  ____________________________________
=  191. 32.112. 0     10111111.00100000.01110000.00000000
```

  - Identifiant d'hôte *(ou plutôt d'interface réseau)* dans un réseau

```
   191. 32.127. 15    10111111.00100000.01111111.00001111
∧    0.  0. 15.255    00000000.00000000.00001111.11111111
  ___________________  ____________________________________
=    0.  0. 15. 15    00000000.00000000.00001111.00001111
```


Fig 4.9 - Exemple 2 de calcul des identifiants réseau et hôte d'une adresse IPv4

Masque naturel

- Si aucune indication de masque d'adresse n'est donnée pour une adresse IPv4, on utilise Le **masque naturel** ou **masque d'adresse par défaut**.
- Le masque naturel est déterminé en fonction d'une ancienne subdivision de l'espace d'adressage d'IP v4 en **5 classes d'adresse**

Classe	Bits de départ	Début	Fin	Notation CIDR	Masque de sous-réseau par défaut
Classe A	0	0.0.0.0	127.255.255.255	/8	255.0.0.0
Classe B	10	128.0.0.0	191.255.255.255	/16	255.255.0.0
Classe C	110	192.0.0.0	223.255.255.255	/24	255.255.255.0
Classe D (multicast)	1110	224.0.0.0	239.255.255.255	/8	non défini
Classe E (réservée)	1111	240.0.0.0	255.255.255.255		non défini

Fig 4.10 - Tableau de 5 classes d'adresse

Masque de sous-réseau

- *Subnetting* : technique qui consiste à diviser un réseau plus large en **plusieurs sous-réseaux**
- Pour subdiviser un réseau en sous-réseaux (*subnet*), on applique **un autre masque d'adresse**.
- On divise en fait la partie « identifiant d'hôte » en 2 parties :
  - un identifiant de sous-réseau (*Subnet number*)
  - un identifiant d'hôte sur un sous-réseau (*Host number*)

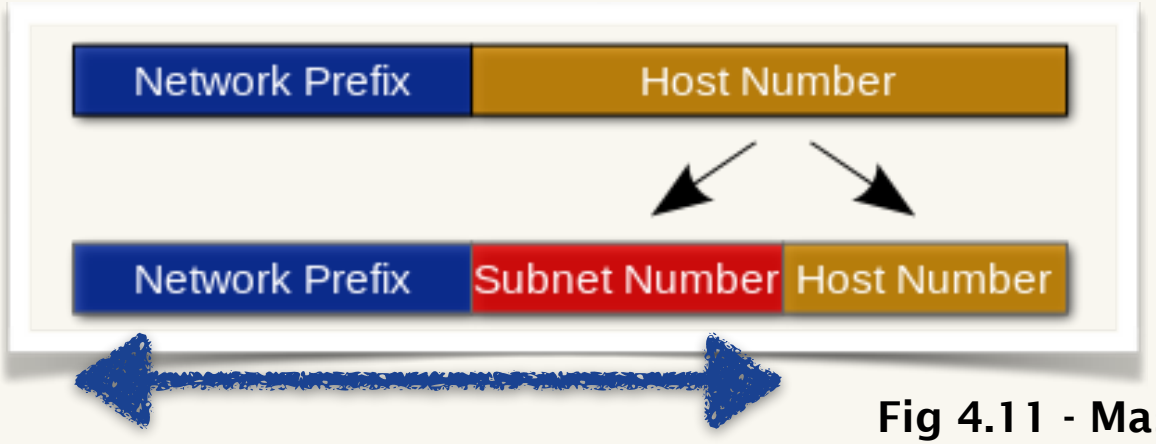


Fig 4.11 - Masque de sous-réseau

- Le masque d'adresse local, ou masque de sous-réseau s'étend jusqu'au champ « **identifiant de sous-réseau** »

Exemples

- Exemple 1
  - Une entreprise dispose du bloc d'adresse : 197.201.30.0 / 24
    - Le masque associé est donc : 255.255.255.0
  - On souhaite créer 16 sous-réseaux :
    - Le masque de sous-réseau voit sa longueur augmenter de 4 (car $2^4 = 16$)
    - ce masque devient donc 255.255.255.240
      - car 240 ≡ `11110000` b
      - (on peut écrire « 255.255.255.11110000 » si cela facilite les calculs)
  - Le bloc devient : 197.201.30.0 / 28 pour le premier sous-réseau
    - Les trois premiers octets sont ceux de l'adresse réseau : 197.201.30.0
    - les quatre bits de poids forts du 4e octet donne l'adresse de sous-réseau
    - les quatre bits de poids faible du 4e octet donne l'identifiant d'hôte dans le sous-réseau

Exemples

- (Suite exemple 1)
  - Le premier sous-réseau, 197.201.30.0 / 28, peut contenir les hôtes d'adresses 197.201.30.1 à 197.201.30.14
  - Il est en effet d'usage d'exclure 197.201.30.0 (qui identifie ce 1er sous-réseau) et 197.201.30.15, qui est utilisé pour une diffusion dans ce sous-réseau.
  - Comment s'écrit le 2e sous-réseau ?
    - ➡ 197.201.30.16 / 28

Exemples

- Exemple 2
  - Un datagramme IP a pour champs d'adresse IPv4 :
    - Adresse source = 193.49.66.200
    - Adresse destination = 193.49.66.29
  - Où se trouvent les ordinateurs source (S) et destination (D) par rapport :
    - Au réseau de préfixe /24 ?
    - Aux sous-réseaux de préfixe /28 ?
  - ➡ S et D sont sur le même réseau 193.49.66.0 / 24
    - Le masque d'adresse est 255.255.255.0
    - Pour S comme pour D, l'identifiant réseau vaut 193.49.66.0

Exemples

- (Suite exemple 2)

	Adresse IP	Dernier octet en binaire
	_ _ _ _ _ _ _ _ _ _ _ _	_ _ _ _ _ _ _ _
S	193. 49. 66.200	1100'1000
D	193. 49. 66. 29	0001'1101
M	255.255.255.240	1111'0000
S∧M	193. 49. 66.192	1100'0000
D∧M	193. 49. 66. 16	0001'0000

- S est sur le sous-réseau 193.49.66.192 / 28
- D est sur un autre sous-réseau 193.49.66.16 / 28

Exemples

- (Suite exemple 2)
  - On peut ausi utiliser une notation mixte 'décimal.binaire' :
  -

	Adresse IP	Notation mixte
S	193. 49. 66.200	193. 49. 66.1100'1000
D	193. 49. 66. 29	193. 49. 66.0001'1101
M	255.255.255.240	255.255.255.1111'0000
S∧M	193. 49. 66.192	193. 49. 66.1100'0000
D∧M	193. 49. 66. 16	193. 49. 66.0001'0000

  - ➡ S est sur le sous-réseau 193.49.66.192 / 28
  - ➡ D est sur un autre sous-réseau 193.49.66.16 / 28

Agrégation d'adresses - CIDR

- Face à la taille croissante d'Internet, **CIDR**, *Classless Inter-Domain Routing* est mis au point dès 1993 afin de réduire les tailles de table de routage.
- **CIDR** (qui se prononce '*cider*') rend la notion de classe d'adresse obsolète
- L'idée est de :
  - permettre le découpage de l'espace d'adressage en **blocs de taille variable**
  - distribuer les blocs d'adresses contiguës à des gros FAI (Fournisseur d'accès à Internet = ISP = *Internet Service Provider*) et en tenant compte de la topologie du réseau
  - modifier les protocoles de routage pour qu'une adresse IP soit accompagnée de la longueur du préfixe associée (VLSM : *Variable-Length Subnet Mask*)

Adresses particulières

- **0.0.0.0 :**
  - soit la route par défaut des tables de routage (la passerelle par défaut)
  - soit « cet hôte » (au démarrage d'une station)
- **Id_réseau à 0** : ce réseau local. Ex. 0.0.0.108 => l'hôte #108 sur ce réseau local
- **Id_hôte tout à 1** : diffusion sur ce réseau
- **255.255.255.255** : *Broadcast* (diffusion générale) sur le réseau local
- **127.0.0.0/8** : test de bouclage. On utilise plutôt 127.0.0.1 (≃ ***localhost***)
- **169.254.0.0/16** : adresses locales auto-configurées (RFC 3927)
  - APIPA (*Automatic Private Internet Protocol Addressing*)
  - Lorsqu'il n'y a pas de serveur DHCP

Adressage privé

- Le RFC 1918 réserve des plages d'adresses à usage **privé**
  - **Non routées** sur internet
  - Au sein de réseaux locaux

Préfixe	Plage d'adresse	Nombre d'adresses
10.0.0.0 / 8	10.0.0.0 – 10.255.255.255	16 777 216
172.16.0.0 /12	172.16.0.0 – 172.31.255.255	1 048 576
192.168.0.0 /16	192.168.0.0 – 192.168.255.255	65 536

- Pour relier un réseau privé à l'Internet, on utilise NAT (*Network address translation*), généralement intégré à un routeur, ou NAPT (*Network address & port translation*)

NAPT (*Network Address and Port Translation*)

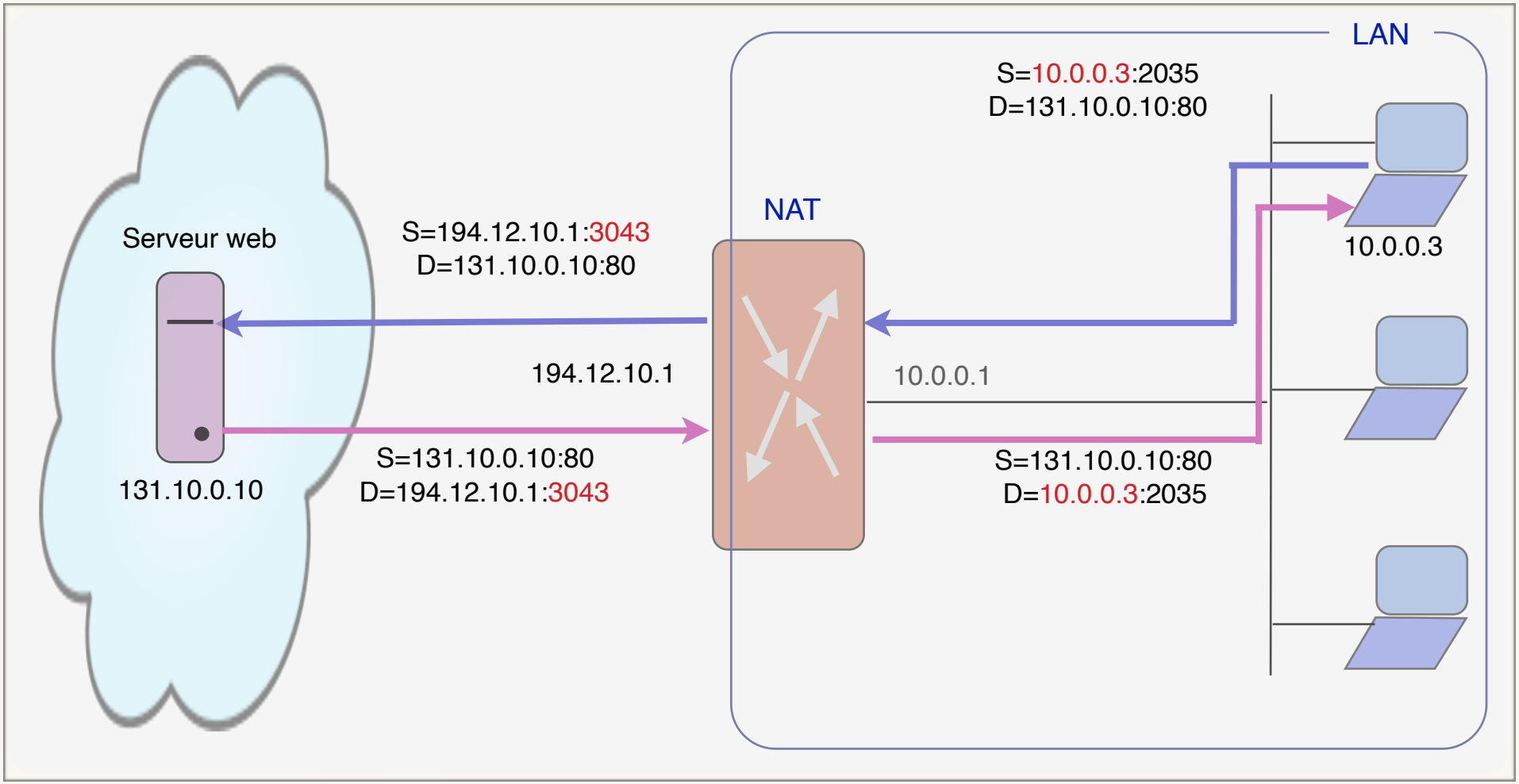


Fig 4.12 -Fonction NAPT d'un routeur

Configuration dynamique

- La **configuration dynamique** d'interface réseau est la plus simple
  - L'ensemble des paramètres IP est délivré par un **serveur DHCP** (*Dynamic Host Configuration Protocol*)
  - Ce serveur DHCP peut être associé au routeur de bordure du réseau local

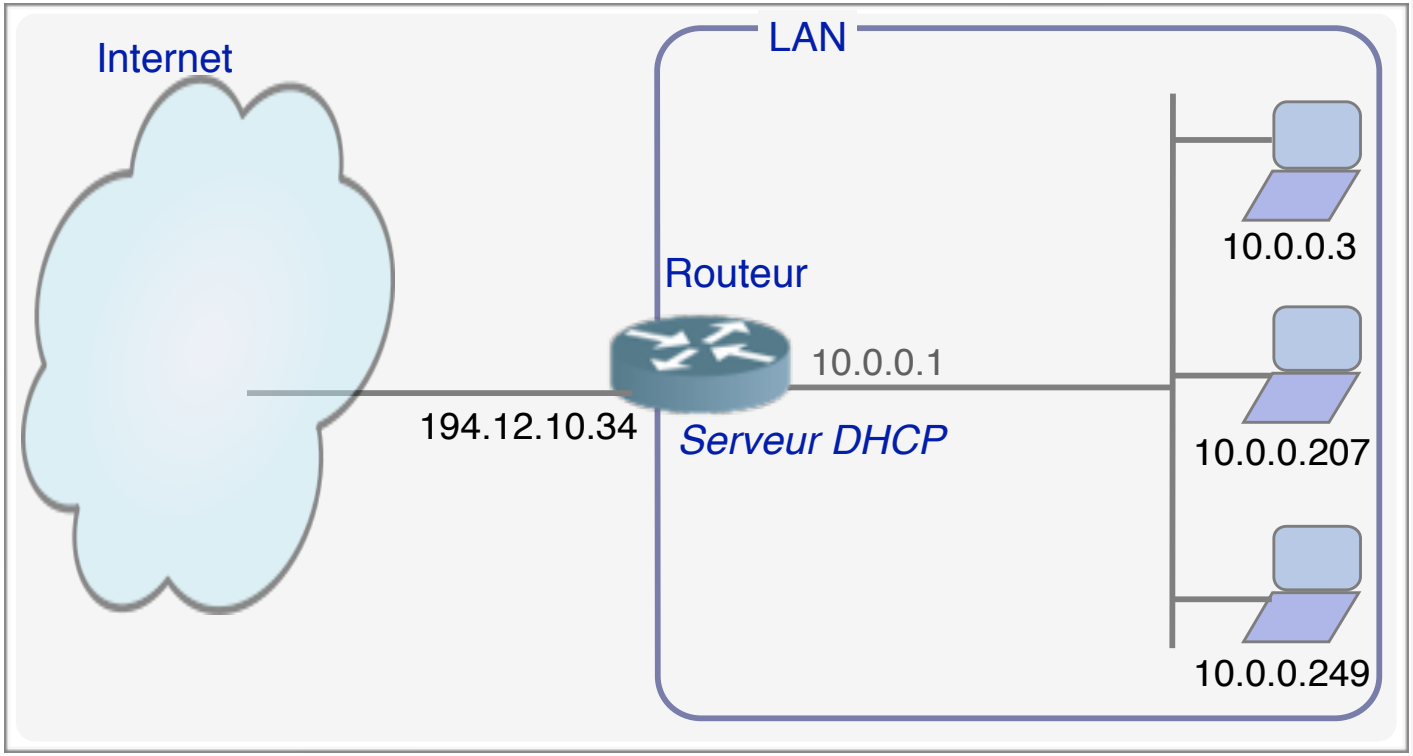


Fig 4.13 - Serveur DHCP associé à un routeur

Configuration dynamique

- Avec Windows 10, 8.1 ou 7
  - Lien : https://support.microsoft.com/fr-fr/windows/modifier-les-paramètres-tcp-ip-bd0a07af-15f5-cd6a-363f-ca2b6f391ace
  - Éditez les propriétés de la connexion (interface réseau) a modifier, puis éditez les propriétés de l'élément « *Protocole Internet version 4 (TCP/IPv4).*

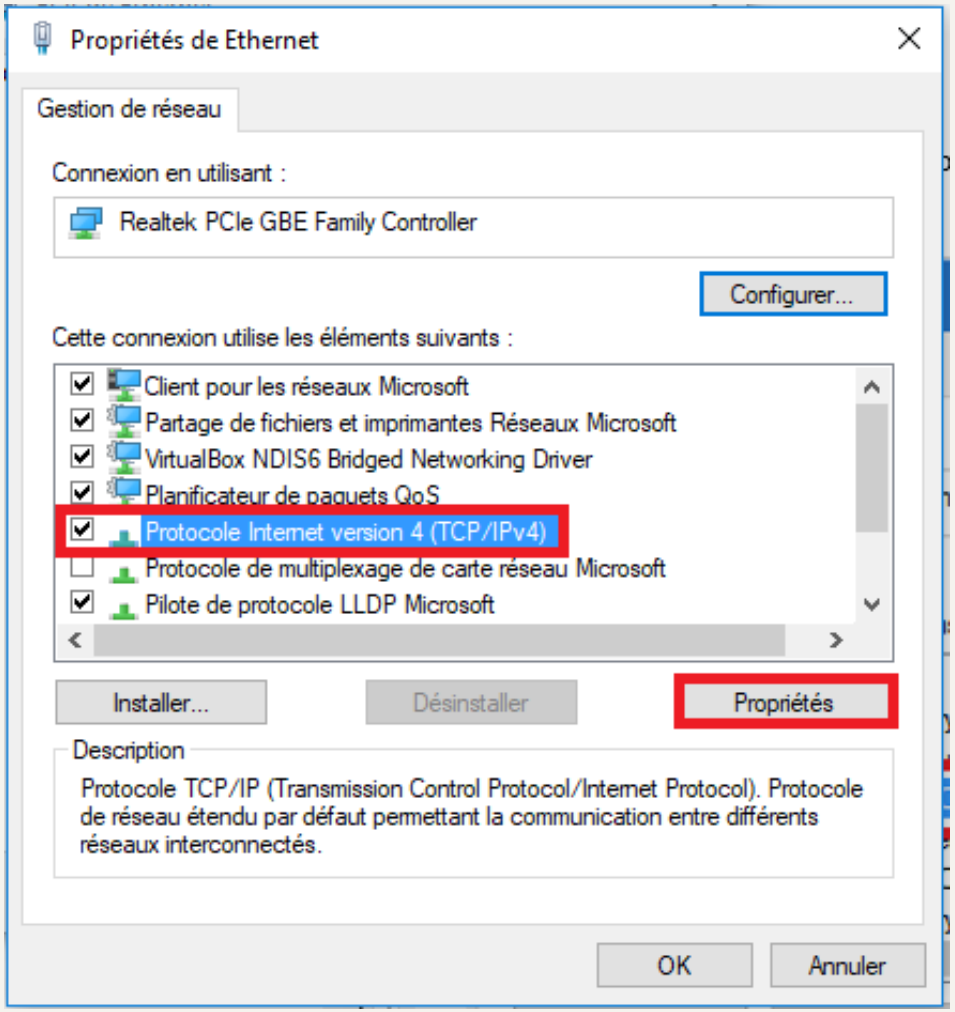


Fig 4.14 - Propriétés de 'Protocole Internet v4' sous Windows

Configuration dynamique

- Avec Windows 11, 10, 8.1 ou 7
  - Lien : https://support.microsoft.com/fr-fr/windows/modifier-les-paramètres-tcp-ip-bd0a07af-15f5-cd6a-363f-ca2b6f391ace
  - Éditez les propriétés de la connexion (interface réseau) a modifier, puis éditez les propriétés de l'élément « *Protocole Internet version 4 (TCP/IPv4).*
  - Cochez « *Obtenir une adresse IP automatiquement* » pour utiliser le protocole DHCP.

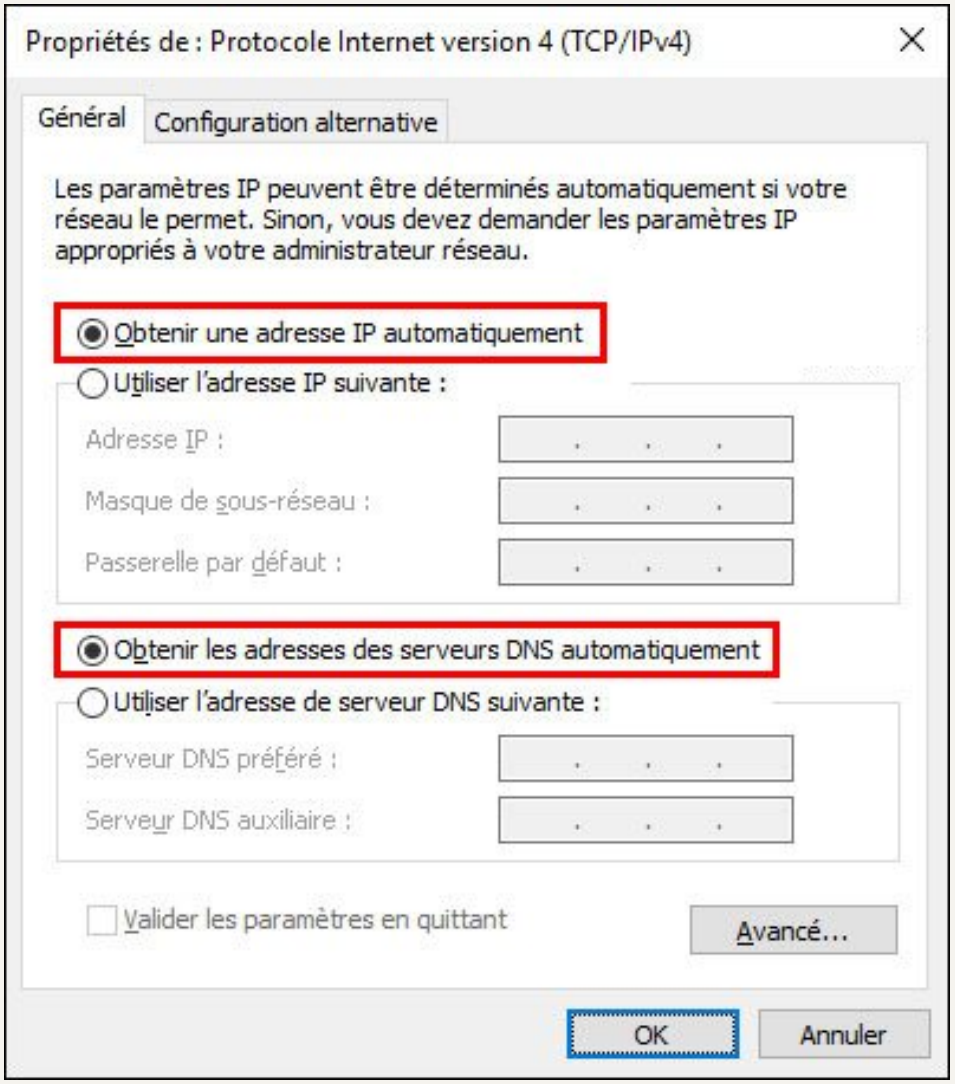


Fig 4.14 - Propriétés de 'Protocole Internet v4' sous Windows

Configuration dynamique

- Avec macOS
  - Lien : support.apple.com/fr-fr/guide/mac-help/mchlp2718/mac
  - Préférences Système / Réseau / Sélectionner un service de connexion
  - Avec le menu déroulant « Configurer IPv4 », choisir « **Via DHCP** »

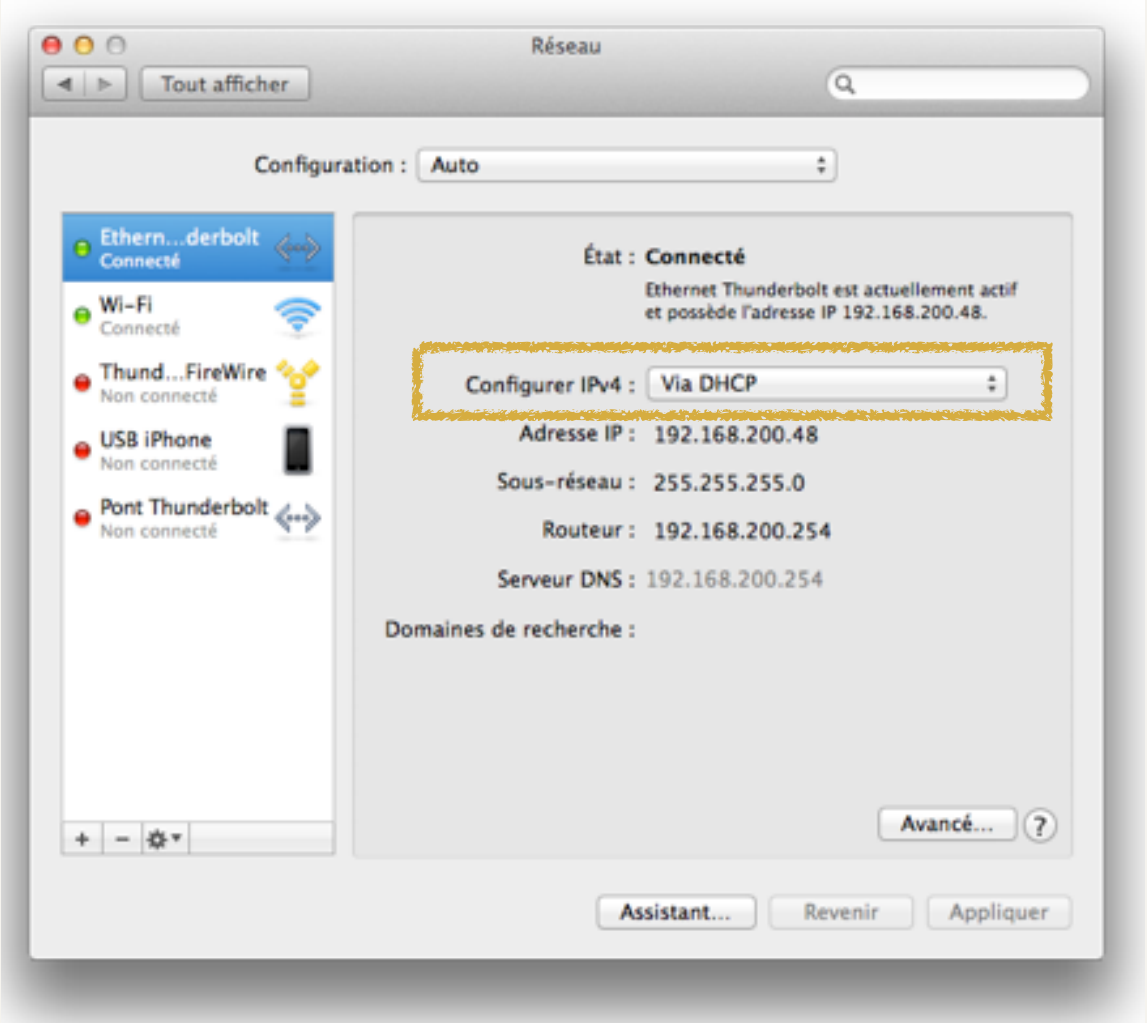


Fig 4.15 - Préférences d'un service de connexion réseau sous macOS

Configuration dynamique

- Avec macOS
  - Lien : support.apple.com/fr-fr/guide/mac-help/mchlp2718/mac
  - Préférences Système / Réseau / Sélectionner un service de connexion
  - Avec le menu déroulant « Configurer IPv4 », choisir « **Via DHCP** »
  - Le bouton « Avancé… » et l'onglet « TCP/IP » donne des réglages avancés

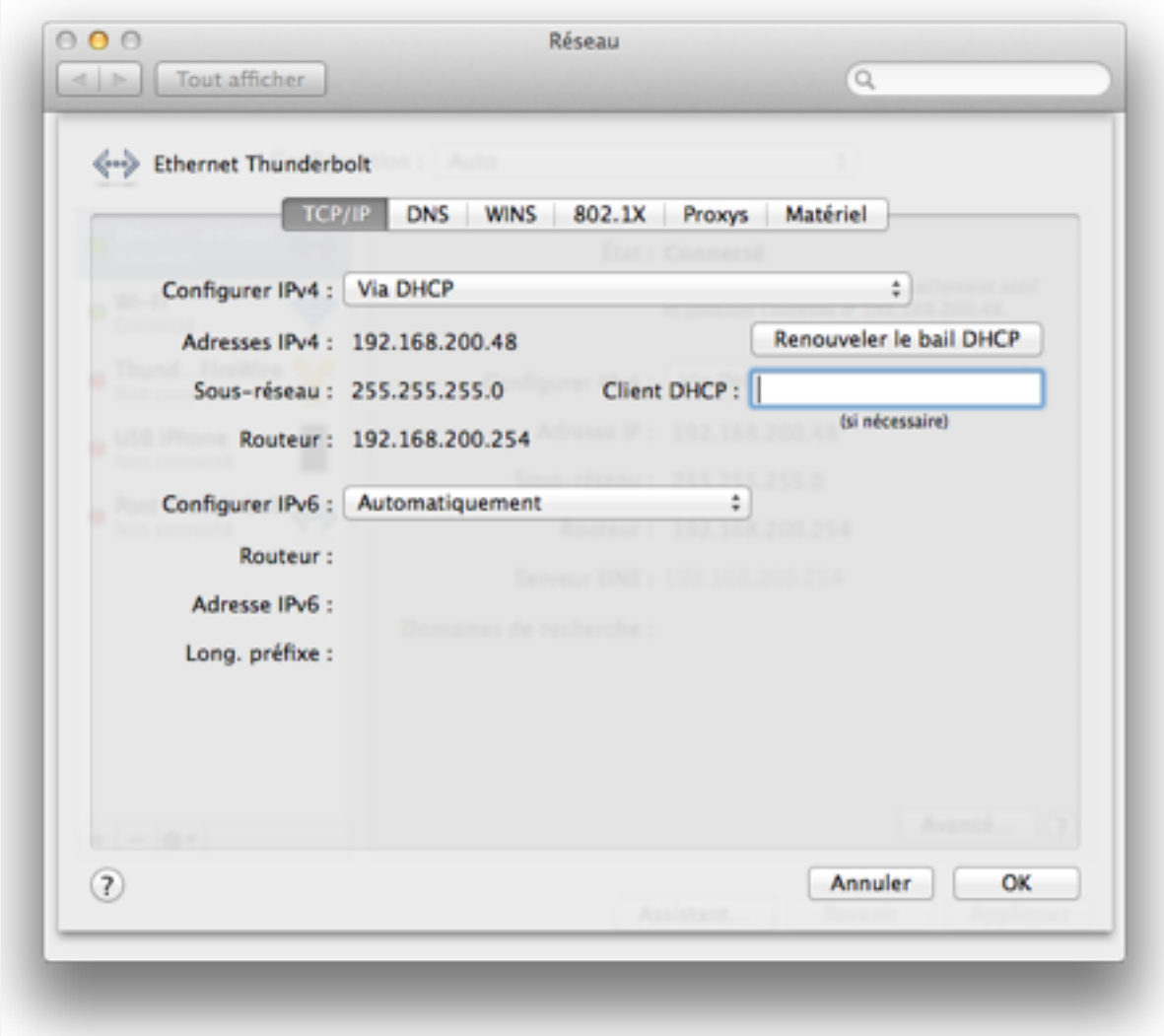


Fig 4.15 - Préférences d'un service de connexion réseau sous OS X

Configuration fixe

- Avec une configuration fixe, les propriétés de TCP/IP (partie adresse IP) sont réglés avec :
  - Une adresse IP
  - Masque de sous-réseau
  - Passerelle par défaut (adresse IP du routeur où seront envoyés les datagrammes hors de portée d'après le masque)

- Nota : d'autres paramètres sont également à renseigner
  - Adresses de serveurs DNS
  - …

Commandes utiles

- Unix et Linux
  - **`ifconfig`**
    - Afficher les informations des interfaces réseau IP actives
  - **`ifconfig -a`**
    - Afficher les informations de toutes les interfaces réseau, actives ou non.

  - La commande **`ifconfig`** est déprécié dans les dernières versions d'Unix. Elle est remplacé par la commande **`ip`**, si le paquet **`iproute2`** est installé.
  - **`ip addr show`** ou son alias **`ip a`**
    - quasi-équivalent à ifconfig

Commandes utiles

- Windows
  - **ipconfig**

```
C:>ipconfig

Configuration IP de Windows

Carte réseau sans fil Connexion réseau sans fil :

   Statut du média. . . . . . . . . . . . : Média déconnecté
   Suffixe DNS propre à la connexion. . . :

Carte Ethernet Connexion au réseau local :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .:
fe80::89c9:7d53:f73:c8c4%11
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.1.2
   Masque de sous-réseau. . . . . . . . . : 255.255.255.0
   Passerelle [[par défaut]]. . . . . . . . . . : 192.168.1.1
```

  - **ipconfig /all**
    - Permet d'avoir toutes les caractéristiques des connexions réseaux : adresse IP, adresse MAC…

ICMP - *Internet Control Message Protocol*

- ICMP est le contrôleur de la couche internet
  - RFC 792
  - Message de contrôle et d'erreur
  - Communications entre routeurs
    - Signalisation de congestion
    - Mise à jour de table de routage
  - Un paquet ICMP est encapsulé dans un datagramme IP.
  - Un champ type de message (8 bits) et un champ Code d'erreur (8 bits) sont principalement utilisé. Exemples :
    - Type : 8 ; Code : 0 => demande d'écho (echo-request)
    - Type : 0 ; Code : 0 => réponse d'écho (echo-reply)
    - Type 11 : Temps dépassé (un TTL a expiré)

ARP - *Address Resolution Protocol*

- ARP permet de trouver l'adresse physique (adresse MAC) d'un hôte à partir d'une adresse IP
  - RFC 826
  - En chaque station, ARP maintient un cache ARP
  - A veut émettre une trame Ethernet vers un ordinateur B d'adresse IP *ad-ip-b* :
  - Si, pour *ad-ip-b*, aucune adresse physique n'est indiqué dans le cache ARP
    - alors A diffuse une requête ARP : Qui est *ad-ip-b* ?
    - seul le poste B avec cet adresse répond : je suis à l'adresse *ad-ip-b* et mon **adresse physique est *ad-mac-b***.
    - A mets à jour son cache ARP
  - A peut donc émettre des trames vers l'adresse physique *ad-mac-b* de B.

Les protocoles de routage

- Les protocoles suivant font partis de la couche Internet de l'architecture TCP/IP. Nous les avons déjà décrit

  - RIP (*Routing Information Protocol*)
  - IS-IS (*Intermediate system to intermediate system*)
  - OSPF (*Open Shortest Path First*)
  - BGP (*Border Gateway Protocol*)

Exemple de table de routage

Network destination ND	Netmask M	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

- *Network destination* et *Netmask* (réseau de destination et masque d'adresse) peuvent être écrit en utilisant la longueur du préfixe :
  - ND = 192.168.0.0 et M = 255.255.255.0 => **192.168.0.0/24**
- Pour savoir si une route ND concorde à une adresse de destination D :
  - **D** ∧ **M == ND** ∧ **M** avec **ND** *Network destination*, **M** *Netmask*
  - Exemple pour 192.168.0.27 :

    192.168.0.27 ∧ **255.255.255.0 == 192.168.0.0** ∧ **255.255.255.0**

Les protocoles de routage

- Plusieurs réseaux de destination (ND) peuvent concorder. Dans ce cas, la meilleure route sera celle où M sera le plus grand (soit **la plus grande longueur** de préfixe).
- Avec cette même table de routage, quelle est la meilleure route pour la destination 192.168.0.100 ?

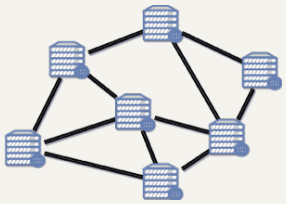
Network destination ND	Netmask M	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

État des lieux

- En prévision d'un manque d'adresses IPv4, l'IETF (*Internet Engineering Task Force*) émet en 1990 un appel à proposition (RFC 1550)
  - 3 propositions sur 21 sont publiées, dont **SIPP** (Simple IP Plus), devenu **IPv6**
  - IPv6 est adopté en 1998 comme successeur d'IPv4
  - RFC 2460

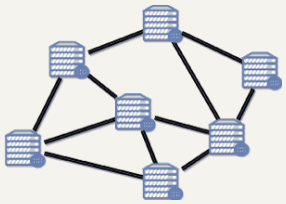
- IPv4, limité à 4 milliard et quelques adresses a pu survivre :
  - malgré une forte demande en Asie et pour les nouvelles applications
  - **grace à** NAT, NAPT (*Network Address & Port Translation*) et CIDR (*Classless Inter-Domain Routing*)

- Les besoins des pays en développement, les équipements mobiles et les objets connectés risquent d'accélérer la pénurie.



Les nouveautés

- Le nombre d'adresses (2^{128} au lieu de ~ 2^{32})
- Les mécanismes de configuration et de re-numérotation automatique
- IPsec, QoS et multicast en natif
- En-têtes simplifiés, cascadables et facilitant le routage



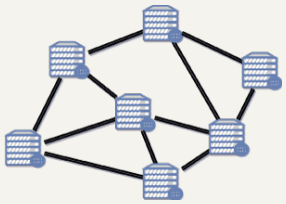
Adresses IPv6

- Une adresse fait 128 bits, soit 16 octets
- On utilise une notation **hexadécimales** :
  - 8 groupes de 2 octets, séparé par « : »
  - Soit 8 groupes de 16 bits ; soit 8 groupes de 4 chiffres hexadécimaux.

- Exemple :
  - **1fff:0000:0a88:85b3:0000:0000:ac1f:8001**

- Il est possible d'exprimer les 32 derniers bits en décimal pointé :
  - **1fff:0000:0a88:85b3:0000:0000:*172.31.128.1***

- Simplifications ou compressions :
  - 1 à 3 **premiers** 0 d'un groupe peuvent être omis
  - **:0000:** <=> **:0:**
  - **:0a88:** <=> **:a88:**



Adresses IPv6

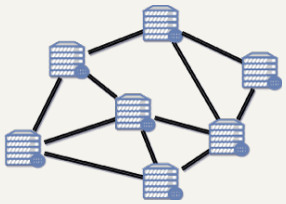
- Simplifications ou compressions (suite) :
  - 1 suite **unique** de groupes de 16 bits tous à 0 peut être omise
  - **:0000:0000:** <=> **::**

- Par exemple :
  - **2001:0db8:39d9:e6c0:0001:0000:0000:0002**

  se simplifie en :
  - **2001:db8:39d9:e6c0:1::2**

- Une adresse IPv4 :
  - **::ffff:193.31.20.46** (correct depuis 2006)
  - **::193.31.20.46** (obsolète)

- Dans une URL, une adresse IPv6 est encadrée par [] :
  - **https://[2001:db8:39d9:e6c0:1::2]/index.php**
  - **https://[2a01:e35:39d9:a1:9e1::b]:8080/index.php**



Sous-réseaux IPv6

- On utilise **le préfixe** et non un masque de sous-réseau
- Exemple :
  - Le sous réseau

    2001:db8:1:1a0:: / 59
  - admet les adresses entre :

    2001:db8:1:1a0::

    et

    2001:db8:1:1bf:ffff:ffff:ffff:ffff
  - car

    59_d = 7x8 + 3 bits

    M = ffff:ffff:ffff:ffe0::

    $a0_h \equiv 1010'0000_b$

    $bf_h \equiv 1011'1111_b$

- Le préfixe **par défaut** a une longueur de 64

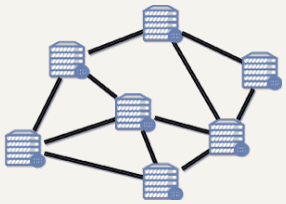
Blocs d'adresses IPv6

- Les adresses **2000::/3** sont les adresses routables sur internet (*Global Unicast*)
- Les autres adresses ne peuvent être utilisées que sur un réseau local, ou par un accord privé de routage mutuel.
- Parmi **2000::/3** on trouve :
  - Le bloc **2001::/16** est ouvert la la réservation depuis 2001
  - Les adresses 6to4 avec le bloc **2002::/16** utilisé pour acheminer du trafic IPv6 sur des réseaux IPv4
  - Les adresses du 6bone avec le bloc **3ffe::/16** utilisé pour une ancienne expérimentation d'interconnexion de réseaux IPv6
  - Les autres sont réservés pour des usages ultérieurs

Blocs d'adresses IPv6

- *Unique Local Addresses :* Les adresses **`fc00::/7`** sont des adresses privées, non routables sur internet (comme 10.0.0.0/8, etc.)
- *Link-Local Addresses :* Les adresses **`fe80::/10`** sont des adresses de lien local, non routables, pour la configuration automatique
  - Les adresses **`fe80::/64`** sont des adresses locales de lien, non routables sur internet, unique sur un lien

- *IPv4-Mapped :* Les adresses **`::ffff:0:0/96`** sont des adresses IPv6 mappant IPv4, pour représenter des adresses IPv4 dans des applications IPv6
- *Loopback :* Adresse de bouclage, alias « localhost » **`::1`**
- *Unspecified :* L'adresse indéterminée **`::/128`** est utilisée pendant l'initialisation d'une machine

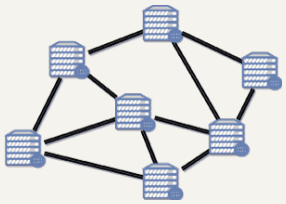


Réservation d'adresses IPv6

- Les adresses IPv6 (comme les adresses IPv4) sont distribuées par blocs par l'**IANA** (*Internet Assigned Numbers Authority*) aux RIR (*Regional Internet Registries*)

- Les RIR sont 5 :
  - **RIPE-NCC** pour l'Europe et le Moyen-Orient
  - ARIN : Amérique du Nord
  - APNIC : Asie & Pacifique
  - LACNIC : Amérique latine et Caraïbes
  - AfriNIC : Afrique
- Un RIR distribue des ressources internet comme les adresses IPv4 et IPv6.

- Les membres du RIPE-NCC sont des LIR (*Local Internet Registries*)
  - Plus de 300 LIR français (opérateurs télécoms, FAI, hébergeurs, etc.)



Utilisation d'IPv6

- Voir **https://6lab.cisco.com/stats/**
- Renater était le premier réseau français à utiliser IPv6
- La majorité des FAI disposent de ressources internes IPv6 et peuvent proposer des connexion IPv6 à leurs clients professionnels
  - La France est assez bien placée au niveau mondial pour le déploiement d'IPv6, malgré une faible motivation des FAI et des opérateurs fixes et mobiles à proposer des offres IPv6 grand public à leurs abonnés.
  - L'Allemagne et des USA sont en bonne progression
- Les systèmes d'exploitation actuels sont compatibles. Voir : fr.wikipedia.org/wiki/Support_de_l%27IPv6_par_système_d%27exploitation

Les atouts d'IPv6

- IPv6 permet principalement de résoudre le problème de la **pénurie d'adresses**
- **Adressage hiérarchique**
  - L'adressage hiérarchique permet de réduire considérablement les entrées dans les tables de routage.
  - En effet, l'agrégation se faisant par opérateur, le nombre de préfixes échangés dans l'Internet entre les différents AS (*Autonomous System*) est donc plus petit.
- **Autoconfiguration**
  - Les stations finales peuvent obtenir une adresse en utilisant le mécanisme d'autoconfiguration qu'intègre IPv6.
  - Ceci allège les tâches d'un administrateur de réseau important.
- **Mobilité et IPsec**
  - La mobilité IP et IPsec sont intégrés nativement dans la pile protocolaire IPv6.
  - Il n'est pas nécessaire de déployer des équipements ou des mécanismes supplémentaires pour les mettre en œuvre.

1 - Préambule

Contenu du chapitre

- *Une lettre ou un appel ?*
  - Transport de données entre un client et un serveur à travers UDP et TCP avec le modèle datagramme, et les approches connecté et non connecté.
    Gestion et utilisation de l'API socket.

2 - Objectifs de la couche transport

Introduction

C'est une **couche clé** du modèle en couche, car à la frontière entre les **fournisseurs** et les **utilisateurs** de la transmission de données

Couche transport	***Client (utilisateur)***
Couche réseau	***Opérateur réseau (transporteur)***

Fig 5.1 - Interface Couche transport / couche réseau

2 - Objectifs de la couche transport

Introduction

Les couches inférieures agissent sur les machines intermédiaires, alors que la couche transport est une vrai couche de bout en bout

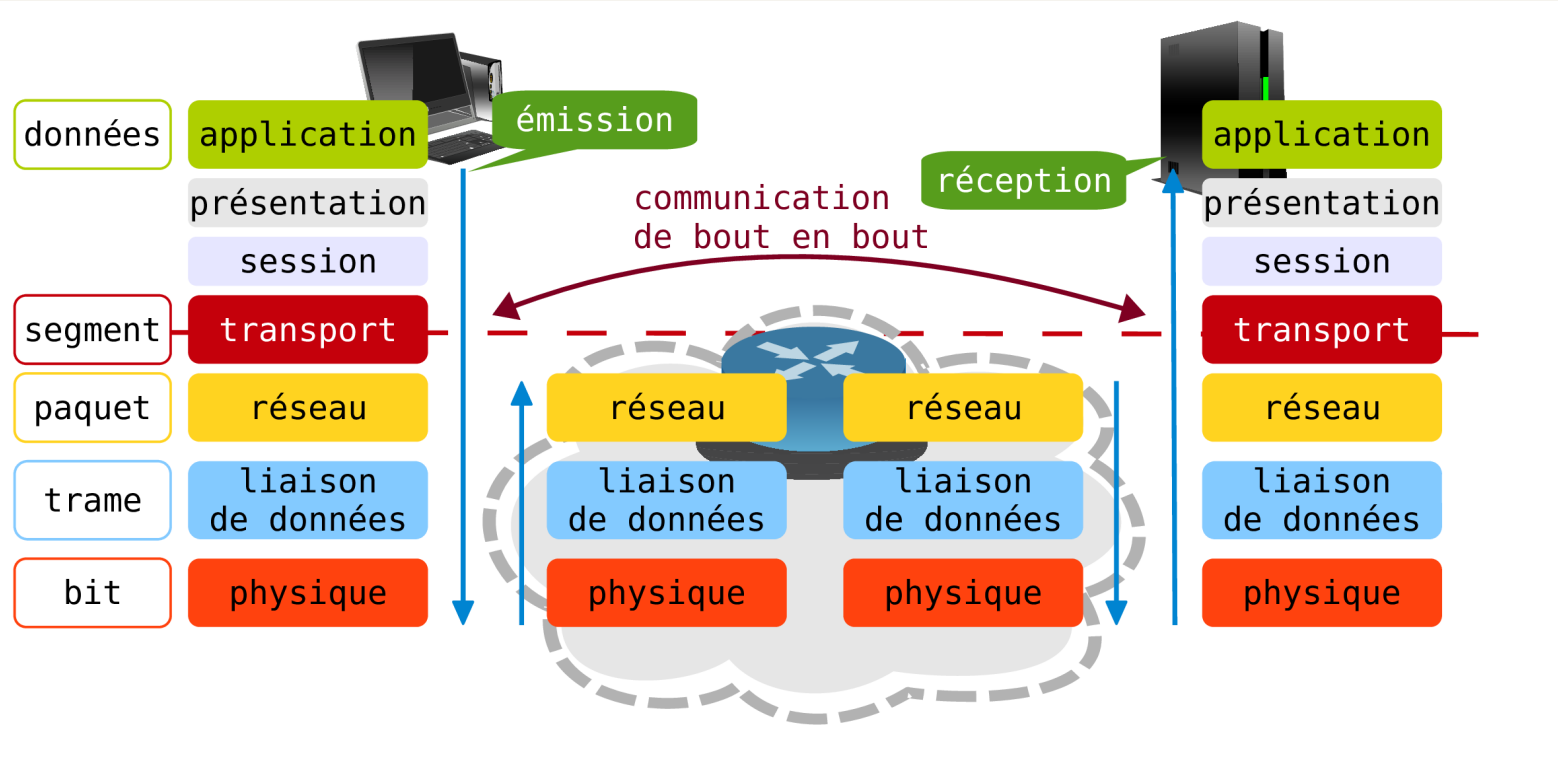


Fig 5.2 - Place de la couche transport dans le modèle OSI

Le service de transport

L'objectif de la couche transport est de fournir à l'utilisateur un service de transport

- efficace
- fiable ou économique
- Indépendamment de la nature du ou des réseaux utilisés

L'entité de transport est la partie logicielle qui assure ces services. Elle est intégrée au système d'exploitation (ou dans un paquetage de bibliothèque).

Types de services

- Services **avec connexion**
- Services **sans connexion**

2 - Objectifs de la couche transport

La qualité de service

QoS (*Quality of Service*) ; elle est liée à différents paramètres :

- temps d'établissement de la connexion
- probabilité d'échec de la connexion
- débit de la liaison
- temps de transit (latence)
- taux d'erreur résiduel
- protection (contre écoutes, intrusions, etc.)
- priorité
- résiliation (probabilité que la couche transport provoque elle même une déconnexion)

Négociation d'options

- L'utilisateur indique les valeurs souhaitées et minimales
- la couche transport propose ces options à la machine distante qui indique ses contre-propositions.

2 - Objectifs de la couche transport

Les primitives de service

Elles permettent l'accès aux services de transport.

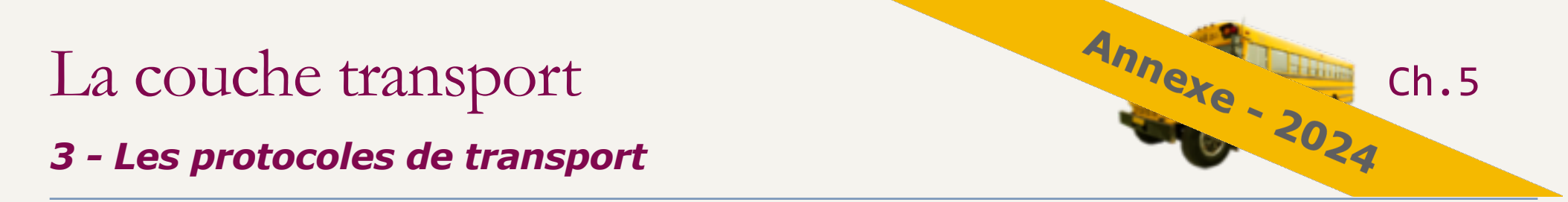
- Il s'agit des primitives d'une API (*Application Programming Interface*)
- Elles sont utilisées par les développeurs d'applications afin de, par ex. :
  - Établir une connexion
  - Utiliser et exploiter cette connexion
  - Libérer la connexion

Exemple de l'interface de connexion par **socket** pour TCP dans l'UNIX de **Berkeley**. Voir par ex.

- Package java.net
- inetdoc.net/pdf/socket-c.pdf

Primitive	Description
`socket`	Créer une nouvelle prise de communication
`bind`	Attacher une adresse locale au socket
`listen`	Annoncer l'acceptation de connexion et la taille de la file d'attente
`accept`	Bloquer l'appelant jusqu'à une tentative de connexion
`connect`	Tenter activement d'établir une connexion sur une socket distante
`send`	Envoyer des données via la connexion
`receive`	Recevoir des données de la connexion
`close`	Fermer la connexion

Fig 5.3 - Primitives de socket



Généralités

Comme pour la liaison de données, le protocole de transport fiable a un rôle de **contrôle** d'**erreurs**, de **séquencement** et de **flux**. Mais l'environnement est très différent.

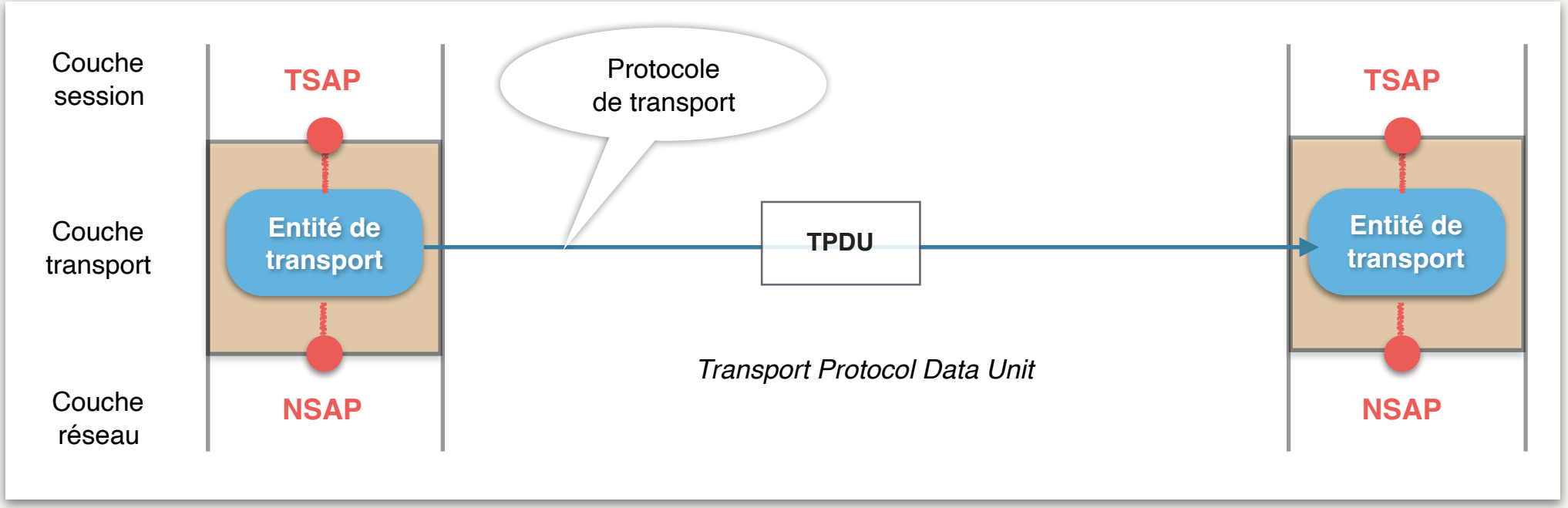


Fig 5.4 - Protocole de transport

Généralités

En liaison de données, l'entité paire est située de l'autre côté de la ligne physique

Pour la couche transport :

- L'adresse de l'entité destinataire doit être spécifié (TSAP)
- Le sous-réseau peut emmagasiner des paquets qui réapparaissent subitement après (mode datagramme)
- La gestion du flux et celle du séquencement seront adaptées au fait que le couche transport gère un nombre important et fluctuant de connexions.

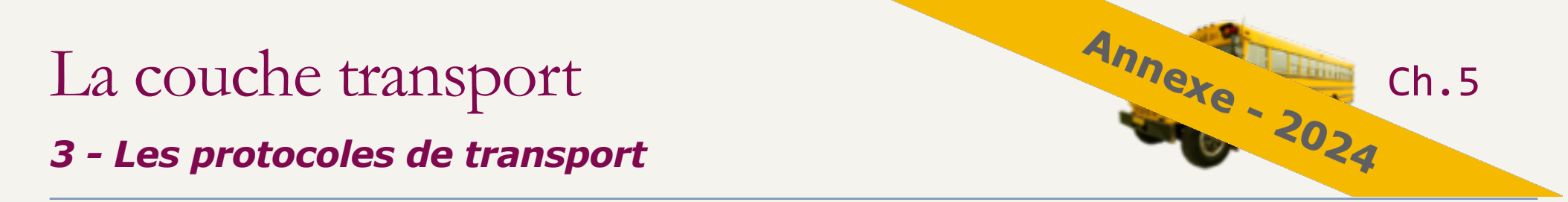
Adressage

Pour l'envoi d'un message ou pour établir une connexion, on doit spécifier à qui s'adresser.
Il s'agit de définir les points d'accès au service de transport

- **TSAP,** *Transport Service Access Point*
- Pour internet, un TSAP est une **adresse de socket** (adresse IP + n° de port)

En général, le TSAP est lié à un service (fourni par un serveur)
Comment le déterminer ?

- Le TSAP est fixe et connu de la source
- La source appelle un annuaire (un service de noms) pour connaitre l'adresse du service à partir de son nom
- Avec un protocole de pré-connexion :
  - Un serveur de processus agit comme délégué (proxy) des serveurs peu utilisées
  - Il écoute un ensemble de ports en attente de connexion
  - Suivant le service demandé par la source lors de la connexion le délégué active le serveur et lui transmet la connexion



Établissement d'une connexion

Une connexion entre deux hôtes est réalisée en **trois étapes** ; la méthode se nomme ***three-way handshake***

- Les numéros de séquence x et y sont choisis de façon aléatoire
- A envoie **CR**, *Connection Request*
- B envoie **Ack**, *Acknowledgement* (accusé de réception)
- A envoie des données

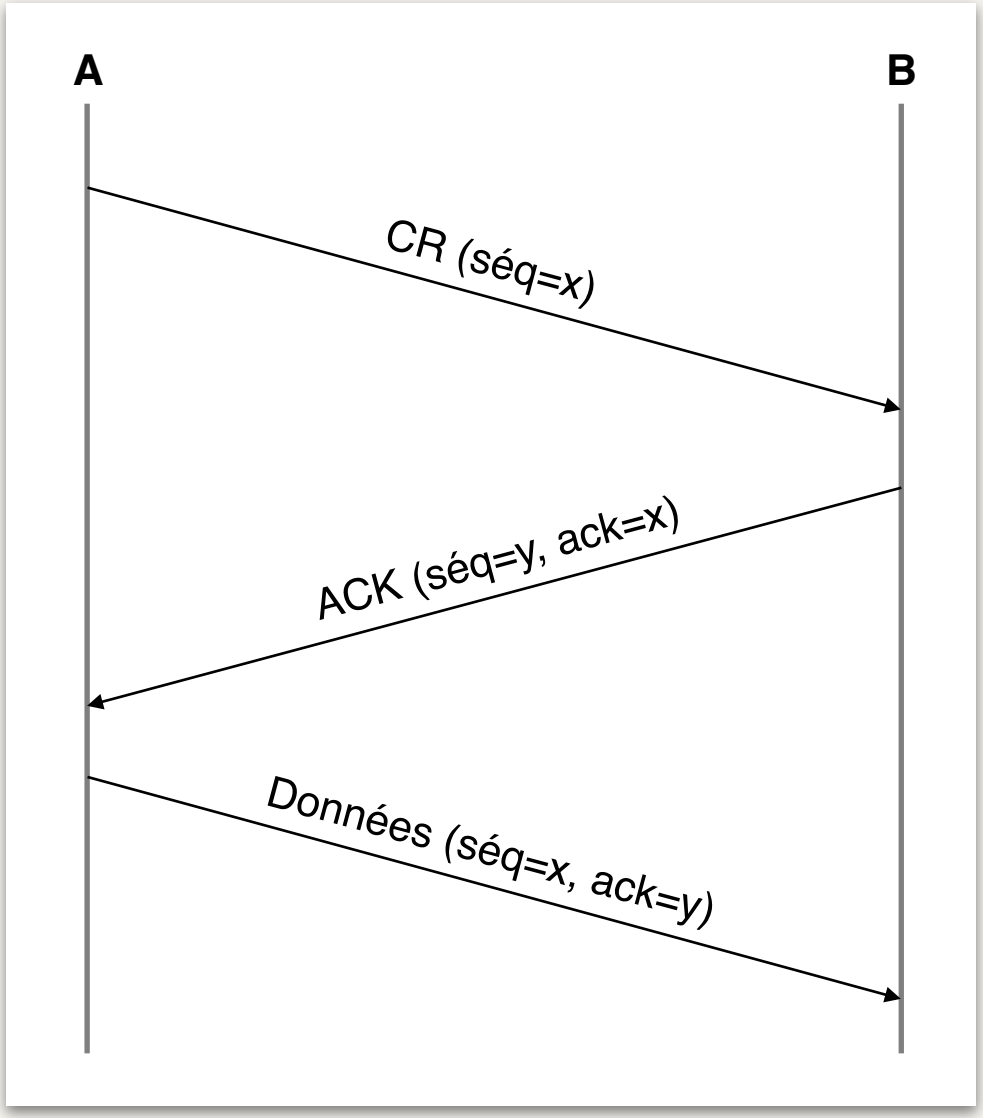


Fig 5.5 - Three-way handshake

Libération d'une connexion

Une libération asymétrique implique un risque de perte de données

On préfère donc une libération symétrique

- Chaque hôte envoi un « *Disconnection request* » et arme un temporisateur
- La figure correspond à une libération normale
- Une déconnexion sera, au pire, réalisé à l'expiration d'un temporisateur (en cas de perte d'un des messages)

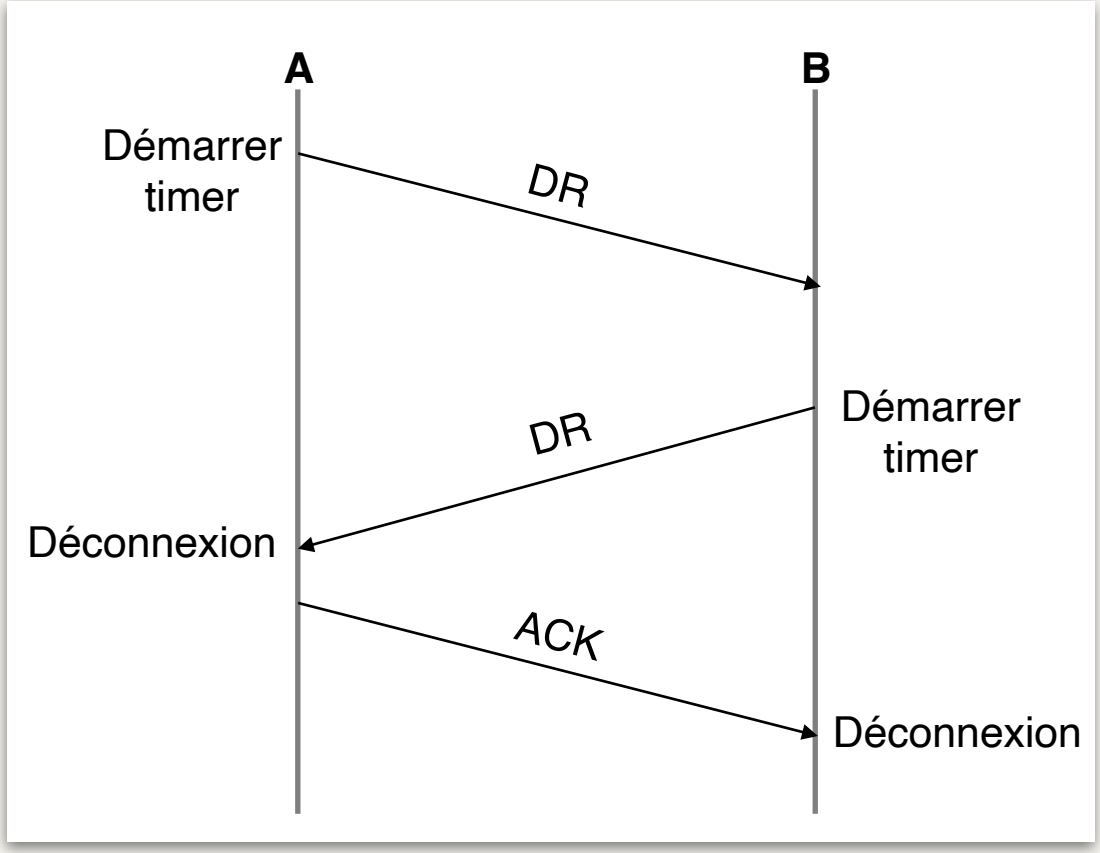


Fig 5.6 - Libération de connexion

Préambule

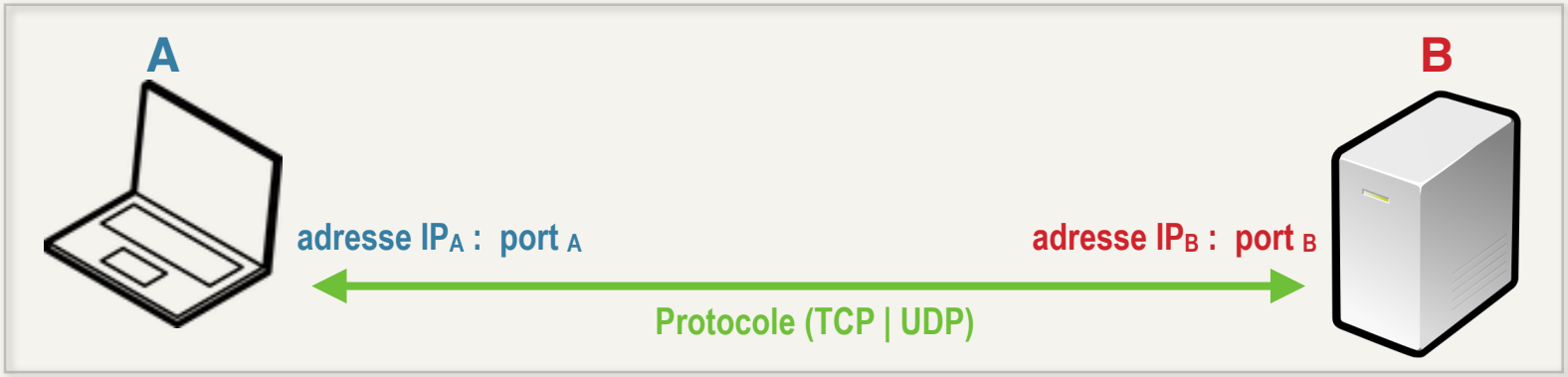
- Les deux principaux protocoles de la couche transport de l'architecture TCP/IP sont :
  - **TCP** (*Transmission Control Protocol*) propose un service fiable orienté connexion
  - **UDP** (*User Datagram Protocol*) est un protocole plus simple, non fiable et sans connexion
- On trouve également :
  - QUIC : nouveau protocole de transport utilisant TLS/SSL et UDP.
  - RTP (*Real-Time Transport Protocol*) permet le transport de données soumises à des contraintes de temps réel, tels que des flux média audio ou vidéo.
    - Il utilise en fait UDP et il permet le transport de média pour les services de la **voix sur IP**, de **vidéo conférence** et de **streaming**.
    - RTP, protocole de transport, est toujours associé à un protocole de niveau Application comme SIP, *Session Initiation Protocol* (VoIP ou vidéo conférence) ou **RTSP**, *Real Time Streaming Protocol*.

TCP - Introduction

- Conçu pour traiter des flots de données de bout-en-bout, de manière fiable, sur un ensemble de réseaux non fiables.
- TCP s'adapte dynamiquement aux variations de paramètres des réseaux (topologie, largeur de bande, délais de transmission, taille de paquets…)
- Résistant aux pannes de la couche réseau et aux pertes de datagrammes.

Le service TCP

- **Notion de socket**
- les connexions TCP sont bidirectionnelles, en mode point à point entre deux adresses de sockets
- Un **socket** est définie par l'association de :
  - Protocole (UDP ou TCP)
  - Adresse IP de la source
  - Port de la source
  - Adresse IP de destination
  - Port de destination
- On peut considérer un **socket** comme une « *prise réseau* »



Le service TCP

- Un numéro de port fait 16 bits
- Certains numéros (< 1024) sont réservés et liés à des services (*Well known ports*)
  - 20 : FTP (data)
  - 21 : FTP (control)
  - 22 : SSH
  - 23 : Telnet
  - 25 : SMTP
  - 80 : HTTP
  - 993 : IMAPS

- Voir le fichier unix **/etc/services**

- Les ports référencés permettent à une application cliente d'identifier une application et un service sur un système distant (un serveur).

Le service TCP

- Groupement de données.
  - La délimitation des messages délivrés par un processus n'est généralement pas conservée.
  - TCP traite des flots d'octets et attends, avant de transmettre des données, que le buffer d'émission soit plein.

  - Dans l'en-tête de TCP un drapeau, PUSH, peut-être utilisé par une application afin que les données soient délivrées immédiatement.
  - Cela est notamment utilisé avec l'action de la touche « Entrée » d'un terminal virtuel.

Le protocole TCP

- Les échanges sont sous forme de **segments**
- Un segment :
  - Un en-tête de 20 octets
  - Un en-tête optionnel
  - Des données optionnelles

- La taille maximum d'un segment est de 65 535 octets (charge utile max. pour IP)

- Les données traitées par TCP sont fragmentées en segment de la taille de MTU (*Maximum Transfer Unit*),
  - unité de transfert maximale
  - caractéristique d'un réseau

L'en-tête TCP

- *source port:* Port source (16 bits)
- *destination port*: Port destination (16 bits)
- *sequence number*: N° de séquence (32 bits) ; n° du premier octet envoyé
- *acknowledgement number*: N° d'accusé de réception (32 bits) ; n° du prochain octet attendu)
- *data offset*: Longueur d'en-tête (4 bits) ; nombre de mots de 32 bits de l'en-tête
- *reserved*: Réservé (6 bits) ; non utilisés

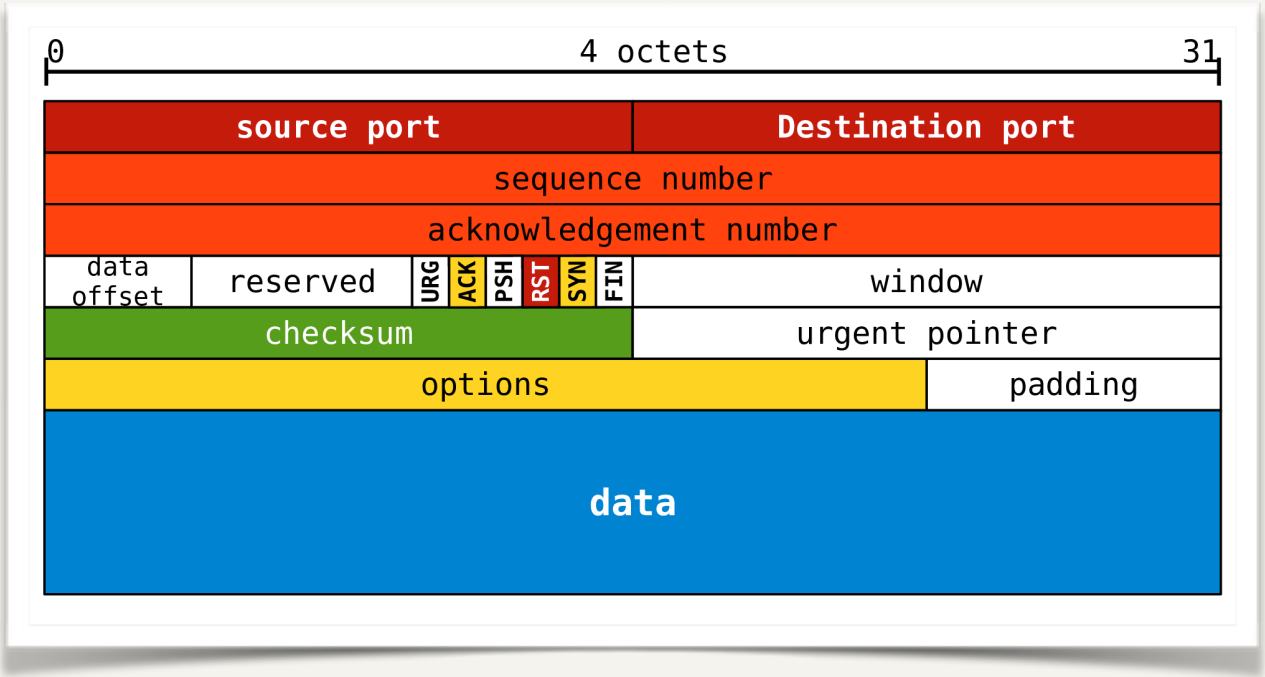


Fig 5.7 - Format d'un segment TCP

L'en-tête TCP

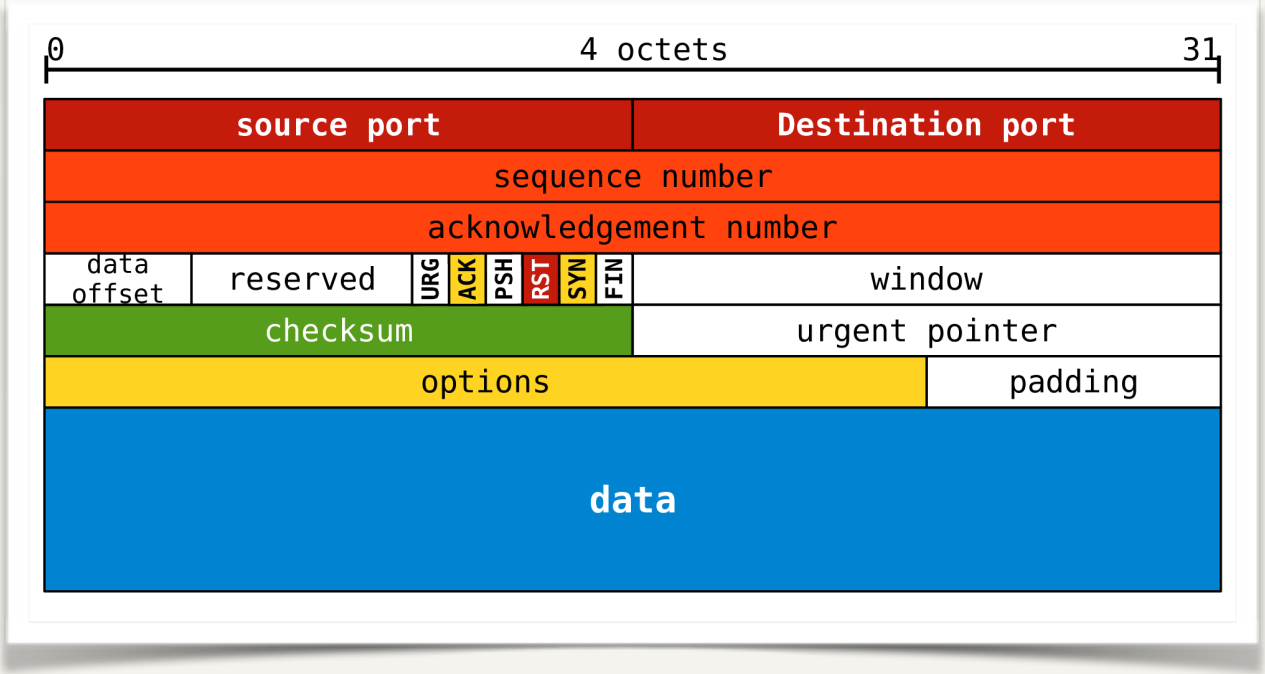


Fig 5.7 - Format d'un segment TCP

- 6 drapeaux d'un bit :
  - **URG**ent
  - **ACK** à 1 si n° d'AR valide (à 0 => AR négatif)
  - **PUSH** ne pas bufferiser les données
  - **RST** à 1 si problème et reset de la connexion nécessaire ou si refus de tentative de connexion
  - **SYN** pour l'établissement de la connexion
    - SYN = 1 et ACK = 0 => CONNECTION REQUEST
    - SYN = 1 et ACK = 1 => CONNECTION ACCEPTED
  - **FIN** à 1 pour libérer la connexion

L'en-tête TCP

- *window*: Taille fenêtre (16 bits) ; taille de la fenêtre d'anticipation = nombre d'octets autorisés à être reçu
- *checksum*: Total de contrôle (16 bits) ; sur l'en-tête, les données et un pseudo en-tête

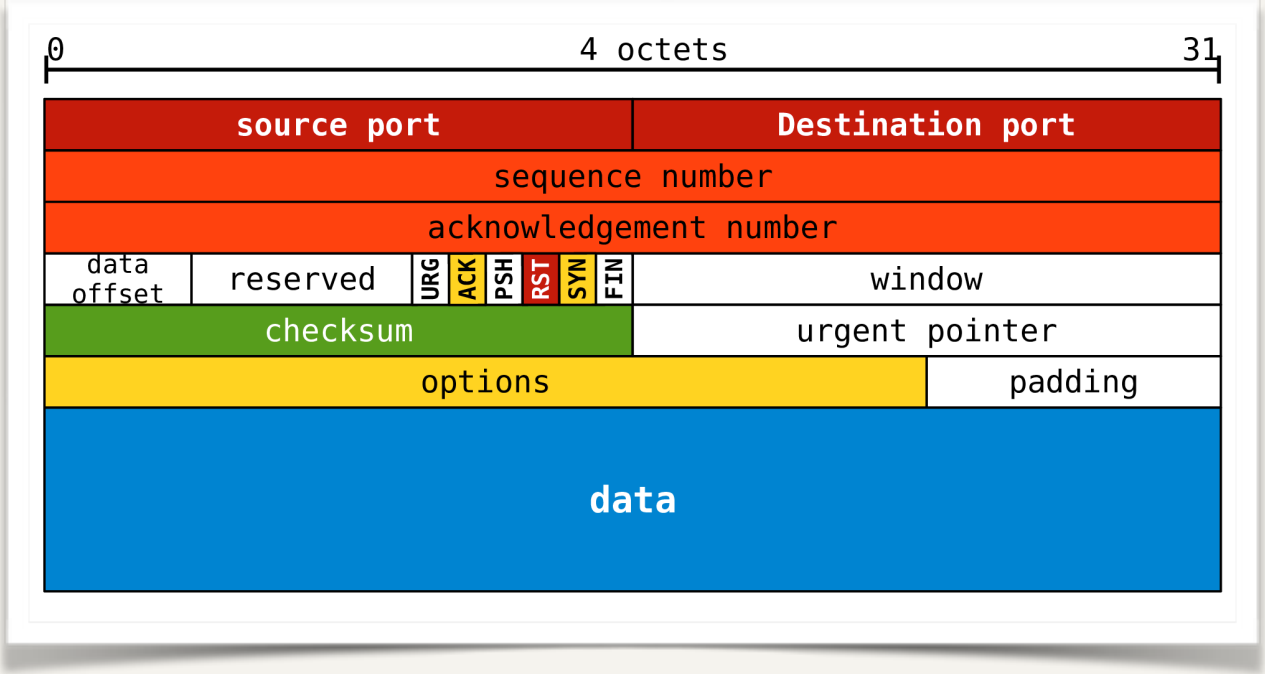


Fig 5.7 - Format d'un segment TCP

- *urgent pointer* : Pointeur d'urgence (16 bits) ; utilisé si URG=1 pour indiquer le décalage en octets par rapport au n° de séquence
- *options* + *padding* : Options + bourrage (0 ou un multiple de 32 bits)
  - Négociation de la taille du MTU en phase de connexion (536 par défaut)
  - Négociation d'un facteur d'échelle de la taille de la fenêtre d'anticipation
  - etc.

Gestion de la connexion TCP

- **Établissement de connexion** avec une poignée de main en **trois étapes** (*Three ways handshake*) : ❶ ❷ ❸
  - Avec la primitive *LISTEN*, le serveur réalise une ouverture passive d'un port
  - ❶ Avec la primitive *CONNECT*, un client réalise demande de **SYNchronisation** au serveur, via une ouverture TCP active, en indiquant :
    - l'**adresse socket** du serveur (@IP + n° de port)
    - le MTU, taille maximum des segments TCP
    - quelques données utilisateurs (ex. identifiant + mot de passe)
    - CONNECT => **SYN=1** ; ACK=0 ; Seq=x

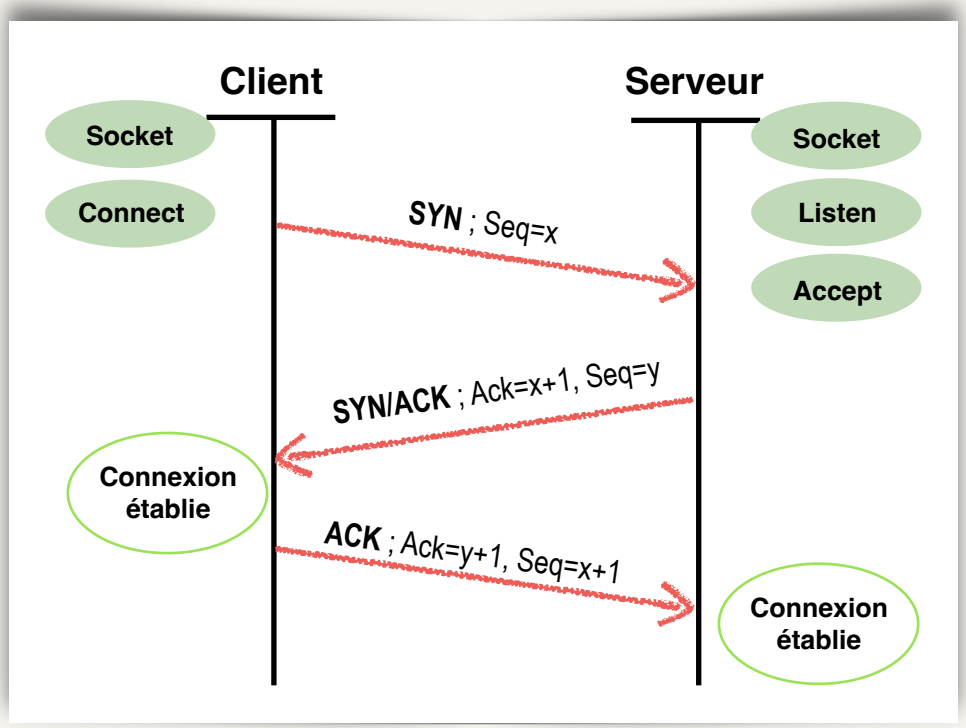


Fig 5.8 - Connexion TCP en 3 étapes

Gestion de la connexion TCP

- Le serveur vérifie que le port demandé est en écoute ; dans ce cas, TCP passe le segment entrant et l'application :
  - ❷ **accepte** la demande de connexion avec la primitive *ACCEPT* par le retour d'un AR avec **SYN=1 ; ACK=1** ; Seq=y
  - ou **refuse** la demande de connexion Rejet => RST=1

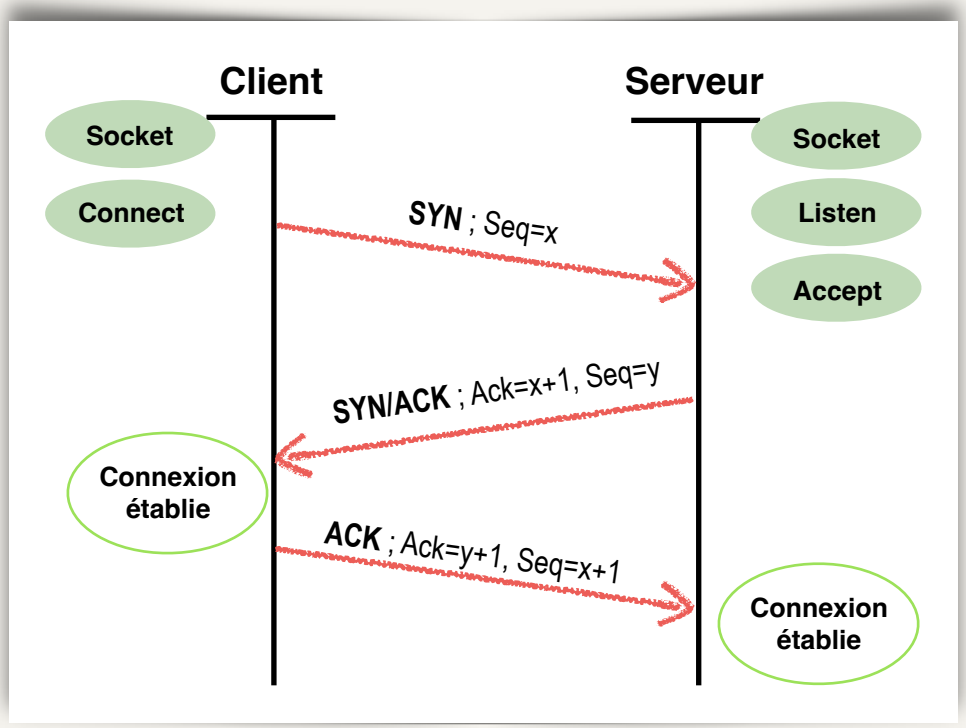


Fig 5.8 - Connexion TCP en 3 étapes

- ❸ Le client qui reçoit SYN/ACK du serveur a sa connexion établie et il retourne un AR :
  - => retour d'un AR avec **ACK=1** ; ack=y+1
- Le serveur reçoit le ACK et la connexion est établie sur le serveur.

Gestion de la connexion TCP

- Si on résume, cette connexion TCP en trois étapes est donc réalisée avec :
  - ❶ **SYN=1**
  - ❷ **SYN=1 ; ACK=1**
  - ❸ **ACK=1**

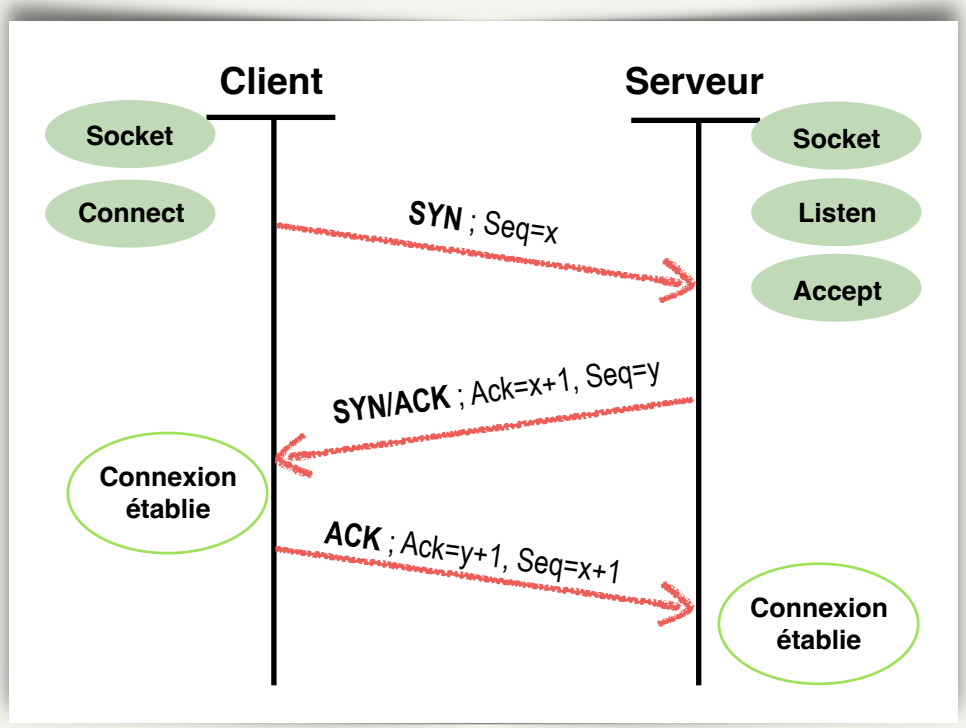


Fig 5.8 - Connexion TCP en 3 étapes

- Voir '*Comprendre les caractéristiques du TCP sous CCNA 200-301 - Vidéo Tuto d'Alphorm*' : https://www.youtube.com/watch?v=lV2B50mTGtM

Gestion de la connexion TCP

- La **libération** de la connexion nécessite **2 FIN** et **2 ACK**
  - Une partie, A, exécute une primitive CLOSE
    - Il envoie FIN à B et arme un temporisateur
  - B reçoit FIN ; il acquitte (ACK=1) et arme un temporisateur
  - B envoie FIN à son tour
  - A acquitte le FIN de B et ferme la connexion
  - B reçoit le ACK de A et ferme la connexion

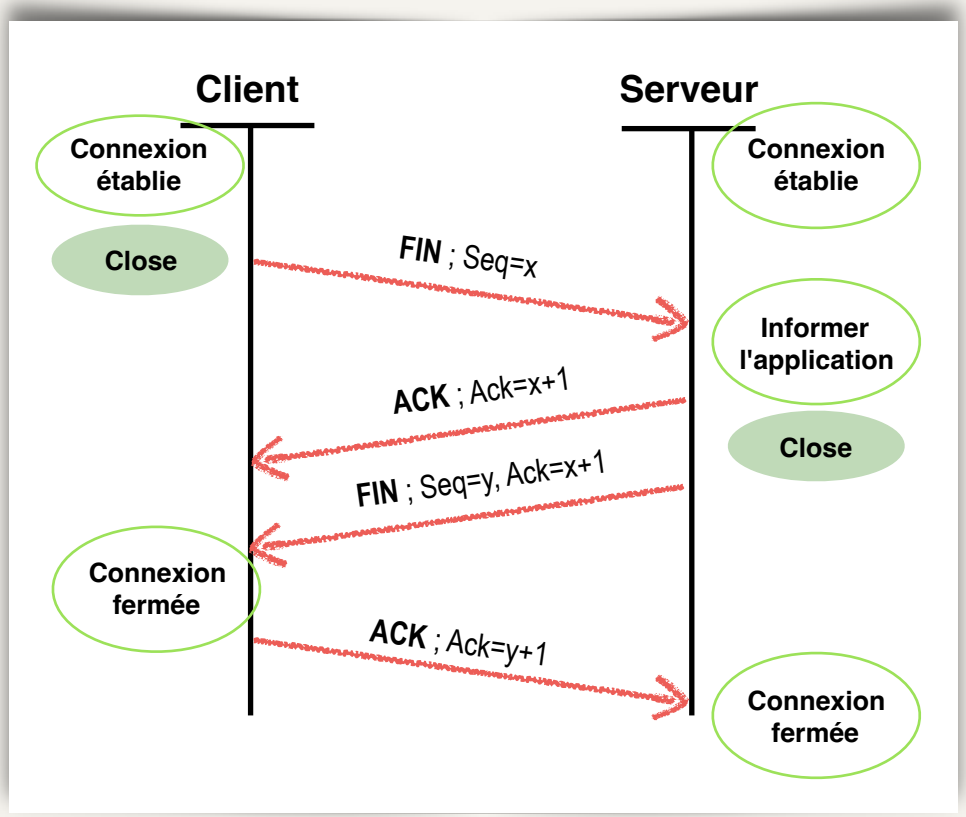


Fig 5.9 - Libération de connexion TCP

- Une des parties qui trouve un temporisateur expiré ferme sa connexion

Fenêtre d'anticipation TCP

- Lié aux **contrôle de flux** et de congestion.
- TCP utilise un mécanisme de fenêtre d'anticipation (ou fenêtre glissante) pour optimiser la communication.
- Cela va résoudre par ex. le problème où le récepteur est plus lent que l'émetteur.
- TCP gère un mécanisme de **fenêtre d'anticipation** de taille variable avec le champ *window*.
- Tant que le champ taille de fenêtre *window* = 0
  - Le **récepteur** indique que son tampon de réception est plein
  - Cela bloque l'émetteur
- Le récepteur indique avec ce champ combien d'octets, *window*, il peut accepter
  - La taille augmente de N_{lu} lorsque l'application a lu N_{lu} octets
  - La taille diminue de $N_{émis}$, nombre d'octet émis par l'émetteur, avec $N_{émis} <= window$.

4 - La couche Transport dans Internet

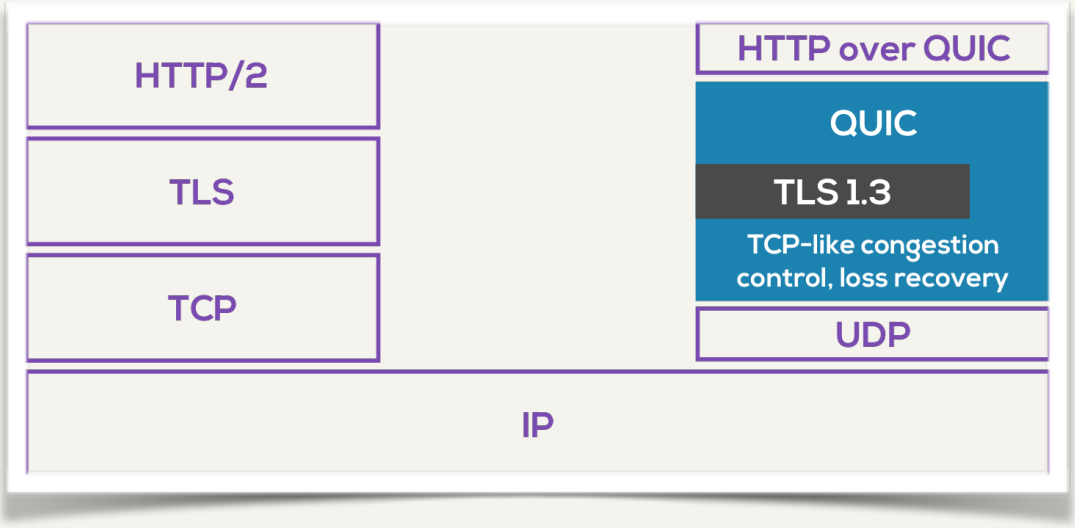
À voir :

- Comprendre les caractéristiques du TCP sous CCNA 200-301 - Vidéo Tuto

UDP : User Datagram Protocol

- Service en mode non connecté
- Les applications utilisent UDP pour encapsuler des données et pour les envoyer sans établir de connexion
- UDP est dédié à des applications :
  - client-serveur ; ex. DNS (*Domain Name System*)
  - en mode dialogue, avec des couples de demande / réponse
  - de streaming (lecture en continu)
  - jeux en réseau…

- QUIC est un protocole de transport qui utilise UDP.
  - QUIC intègre TLS 1.3, *Transport Layer Security* 1.3
  - HTTP/3 s'appuie sur QUIC



UDP : User Datagram Protocol

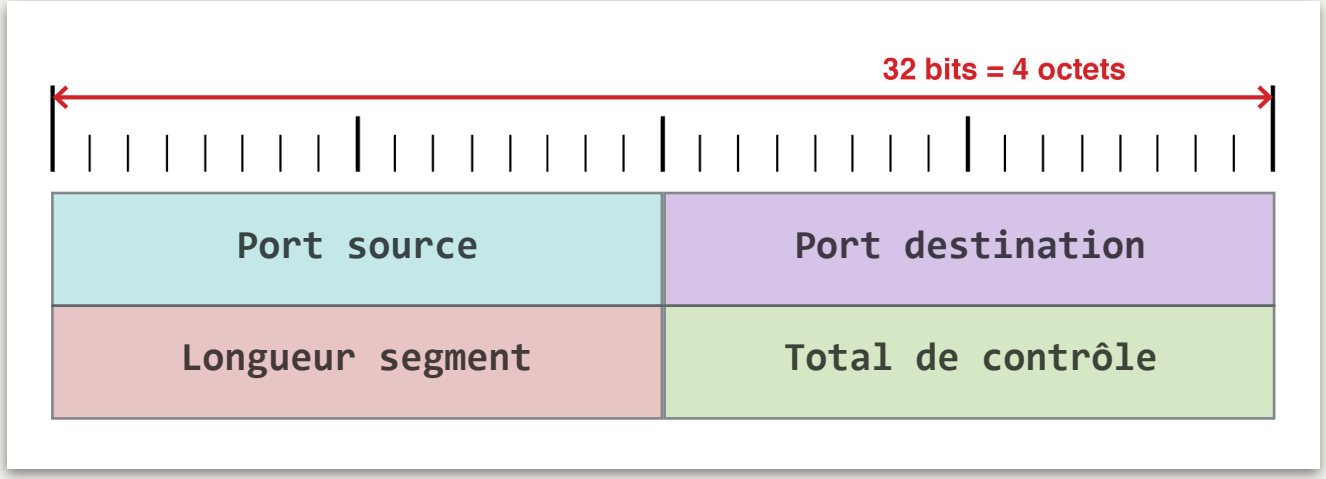


Fig 5.10 - L'en-tête UDP

- En-tête de **8 octets**
- 4 champs de 16 bits
- Port source et destination
  - idem TCP
- Longueur du segment
  en octets, en-tête inclus.
- Total de contrôle (ou 0 s'il n'est pas calculé)
  comme TCP, sur :
  - Pseudo en-tête IP (adresse source, adresse destination, protocole, taille UDP)
  - En-tête UDP
  - Données

Couche application

- Cette couche regroupe les niveaux session, présentation et application du modèle OSI. Elle contient les protocoles de haut-niveau utilisés par les logiciels pour leur besoin de communication.
- Exemples de protocoles :
  - **Transfert de fichiers** : FTP, *File Transfer Protocol* ; SFTP, *Secure File Transfer Protocol*…
  - **Messagerie électronique** : SMTP, *Simple Mail Transfer Protocol* ; POP3, *Post Office Prot.*; IMAP, *Internet Message Access Prot.* ; MIME, *Multipurpose Internet Mail Extensions*…
  - **Messagerie instantanée** : XMPP, *Extensible Messaging and Presence Protocol*, alias *Jabber* ; IRC, *Internet Relay Chat*…
  - **Travaux à distance** : Telnet ; ssh, *Secure shell*
  - **Consultation et gestion d'annuaires :** LDAP, *Lightweight Directory Access Protocol*
  - **Standards et outils du web : HTTP**, *HyperText Transfer Prot.* ; **URI**, *Uniform Resource Identifier* ; **HTML**, *HyperText Markup Language* ; **CGI**, *Common Gateway Interface*…
  - **Traduction de nom de domaine en adresse IP** : **DNS**, *Domain Name System*
  - **Autres** : NFS, *Network File System* ; SNMP, *Simple Network Management Protocol* ; DHCP, *Dynamic Host Configuration Protocol* ; *etc.*

DNS, *Domain Name System*

- DNS, Système de Nom de Domaine, correspond à différentes notions :
  - Une base de données répartie, grâce à un grand nombre de serveurs de noms qui communiquent entre eux.
  - Un protocole, de niveau application, qui utilise UDP pour le transport
    - Le port 53 est utilisé par convention.
  - DNS est un service de nommage standard sur internet (RFC 1033 à 1035).
    - **Un des objectifs** est donc de retrouver l'adresse IP d'une machine à partir de son nom de domaine.

Nom de domaine

- Un nom de domaine est composé d'un domaine de haut-niveau, **TLD**, ***Top Level Domain***, précédé de **un ou plusieurs** sous domaines.
  - On distingue
    - les gTLD, *generic Top Level Domain* (ex.: .com, .net, .org, .paris, .guru…)
    - les ccTLD, *country code Top Level Domain* (ex.: .fr, .uk, .jp, .nl, .eu, .us…)

Nom de domaine, suite…

- Chaque sous-domaine est validé et enregistré auprès du domaine supérieur.
  - Exemple :

paul.info.arcep.fr

fr = ccTLD pour la France ; géré par l'AFNIC

arcep = Domaine de second niveau, enregistré et validé par l'AFNIC (propriétaire : l'Autorité de Régulation des Communications Electroniques et des Postes)

info = Service informatique validé et enregistré pour l'ARCEP.

paul = Nom d'une machine (celle de Paul) au sein du service informatique de l'ARCEP.

Nom de domaine, suite...

- **FQDN**, *Fully Qualified Domain Name* est un nom de domaine écrit de façon absolue. Il se termine par **un point final** qui représente la racine.
  - Exemple : La hiérarchie du domaine de.wikipedia.org.

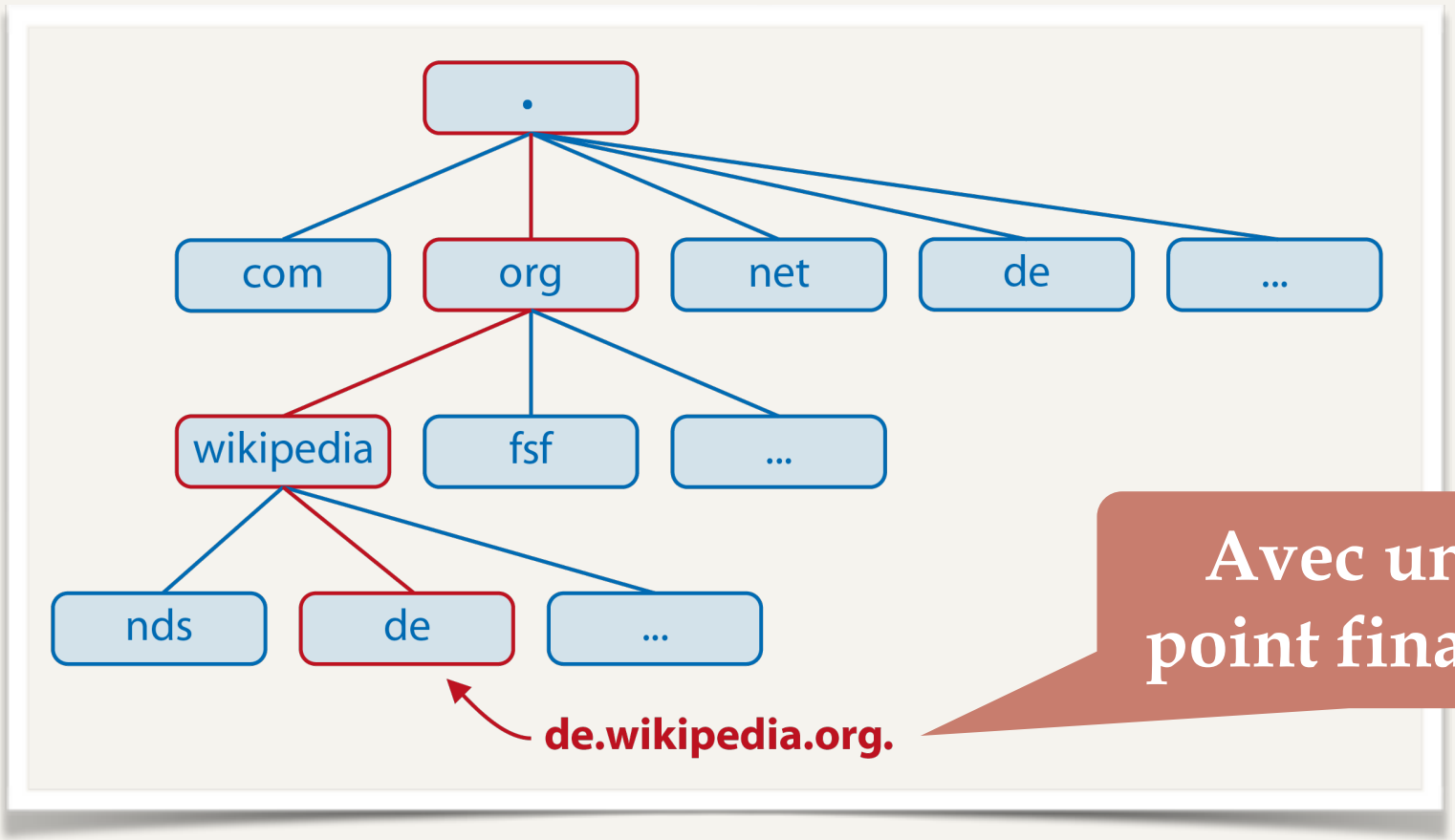
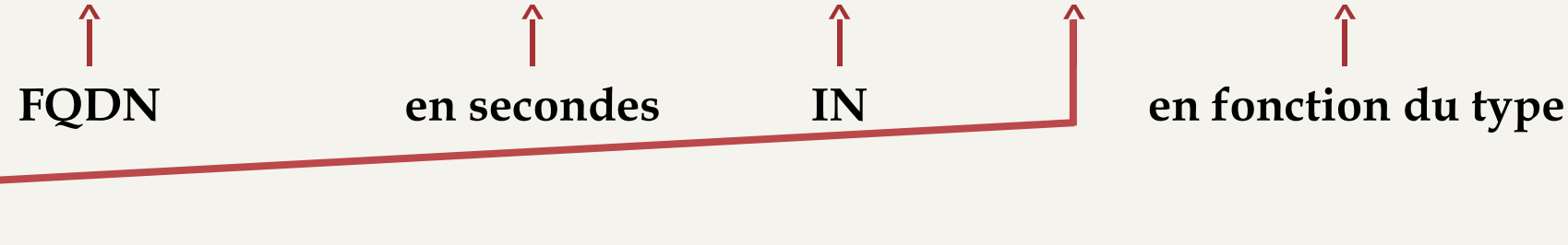


Fig 6.1 - La hiérarchie du FDQN de.wikipedia.org.

Enregistrement de ressources

- Un enregistrement de ressources, ***resource record***, se compose de **cinq éléments** :

```
Nom de domaine ;  Durée de vie ; Classe  ; Type ;    Valeur
```



- A = *Address Record* : la valeur est l'adresse **IPv4** du nom de domaine
- AAAA = *Address Record* : la valeur est l'adresse **IPv6** du nom de domaine
- MX = Relai de messagerie
- SOA = *Start of Authority* : serveur principal d'une zone
- NS = Serveur de noms
- CNAME = nom canonique : Alias de nom de domaine
- PTR = Pointeur
- HINFO = description de l'hôte
- TXT = Texte de commentaire
- …

Enregistrement de ressources, suite

- Exemple :

Nom de domaine	Durée de vie	Classe	Type	Valeur
fr.wikipedia.org.	575	IN	CNAME	text.wikimedia.org.
text.wikimedia.org.	1522	IN	CNAME	text.esams.wikimedia.org.
text.esams.wikimedia.org.	2193	IN	A	91.198.174.232

- Voir les commandes `dig`, `nslookup` et `host`
  - `dig fr.wiktionary.org`
  - `dig mx fr.wiktionary.org`
  - `host fr.wiktionary.org`

La couche application de TCP/IP

3 - HTTP

HTTP, *Hypertext Transfer Protocol*

- Le web ; WWW, *World Wide Web*
  - Créé par Tim Berners-Lee au CERN (Conseil Européen pour la Recherche Nucléaire)
  - Le web repose principalement sur les standards :
    - HTML, *HyperText Markup Language*
    - URL, *Uniform Resource Locator.*
    - **HTTP**, *Hypertext Transfer Protocol*
    - CGI, *Common Gateway Interface*
    - …
- HTTP est :
  - Simple ; les messages sont lisibles
  - Extensible
  - Sans état : pas de lien entre deux requêtes qui sont effectuées

Fig 6.2a - Tim Berners-Lee

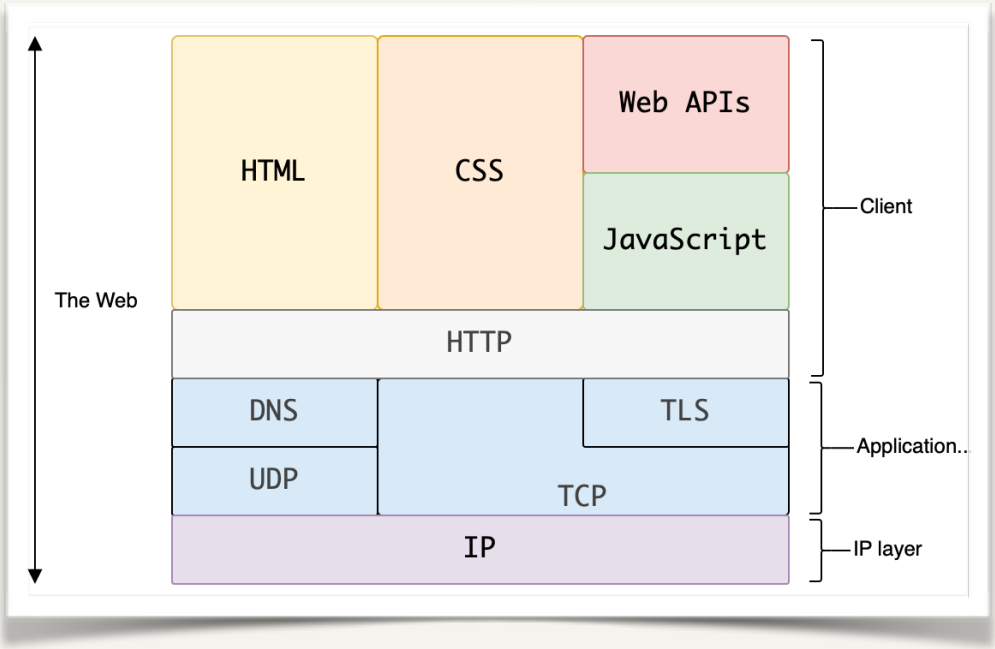


Fig 6.2b - Couches liées au web

https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

HTTP, *Hypertext Transfer Protocol*

- HTTP est un protocole qui permet de récupérer des ressources telles que des documents HTML et des éléments qui composent ces documents.

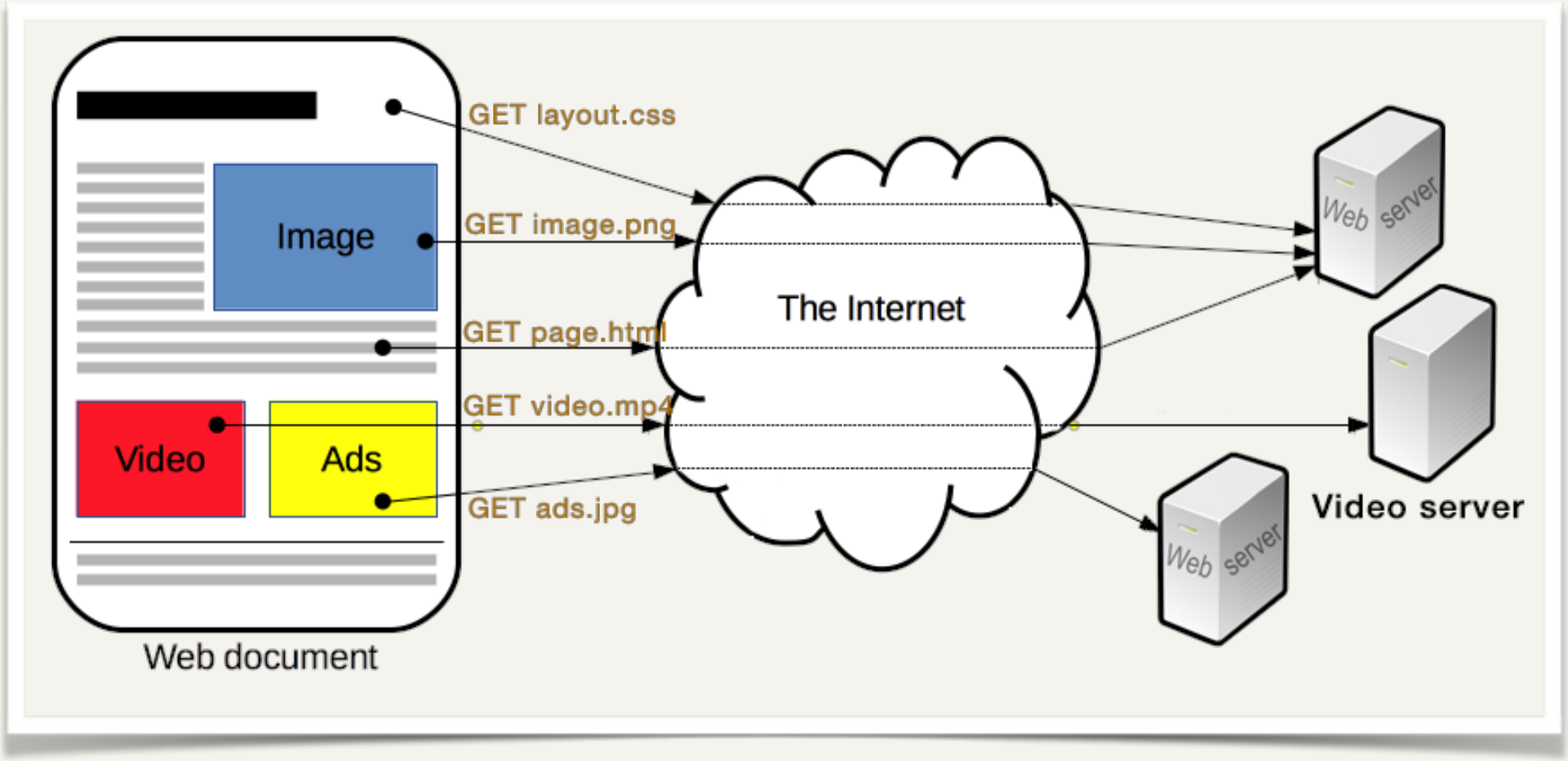


Fig 6.3 - Requêtes pour récupérer des ressources d'un document web

HTTP, *Hypertext Transfer Protocol*

- HTTP est un protocole qui permet de récupérer des ressources telles que des documents HTML.
- À la base de tout échange de données sur le Web.
- Protocole de type client-serveur :
  - les requêtes sont **toujours** initiées par le client (agent utilisateur, ou son proxy), par ex. un navigateur web, un robot web [*bot*], etc.
  - Un document complet est composé de différents sous-documents qui sont récupérés, par exemple :
    - du texte (html),
    - des descriptions de mise en page (css),
    - des images (png, jpeg…),
    - des vidéos (mp4…),
    - des scripts (js)
    - et bien plus.
- Les ressources sont identifiées par leur URL, *Uniform Resource Locator*.
- Les URL et les URN, *Uniform Resource Name* sont des sous-ensembles des URI, *Uniform Resource Identifier*.

HTTP, *Hypertext Transfer Protocol*, suite

- Chaque requête individuelle est envoyée au serveur, qui la traite et fournit une réponse. Le serveur peut être composé de :
  - load balancing : un ensemble de serveurs pour répartir la charge
  - N-Tier Architecture : architecture complexe où le serveur web utilise un serveur d'application, lui même client d'autres serveurs (base de données, etc.)

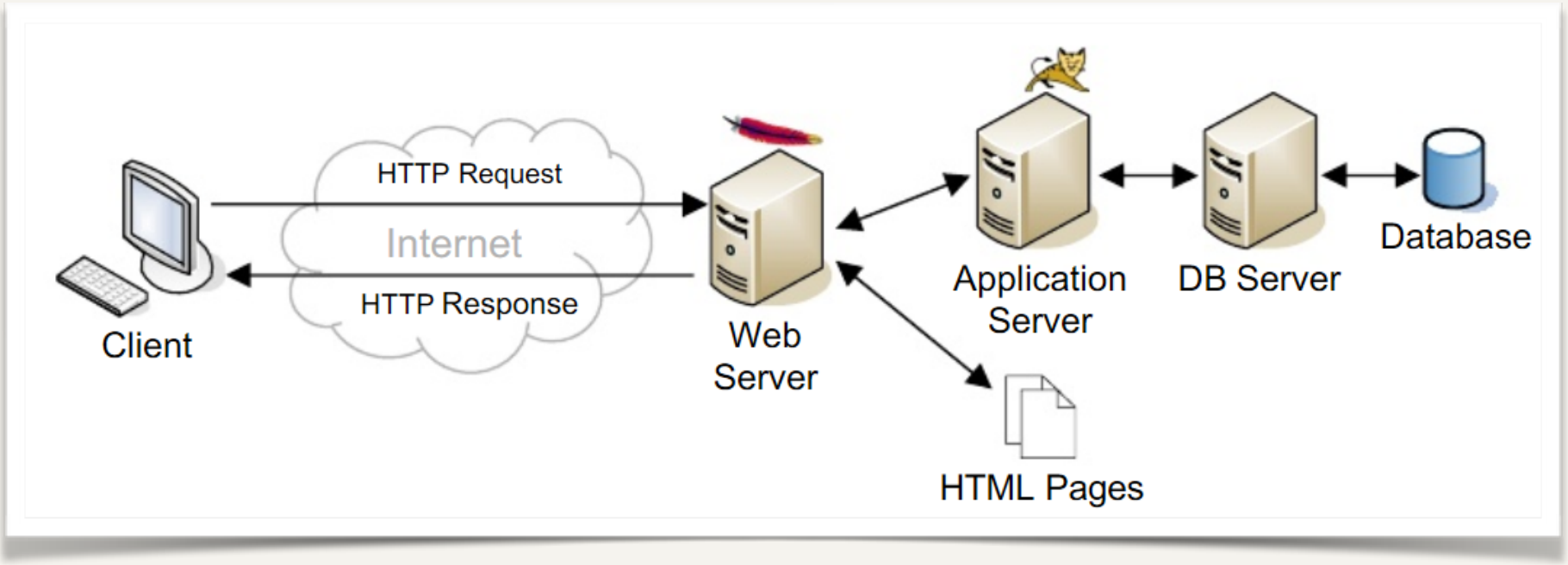


Fig 6.4 - Architecture N-Tier

Flux HTTP

- Lorsqu'un client veut communiquer avec un serveur il réalise les étapes suivantes :
  - Résolution du nom de domaine défini dans l'URL => adresse IP du serveur (@IP_serveur)
  - Ouverture d'une connexion TCP. Par défaut sur **@IP_serveur:80** ou **@IP_serveur:443**
    - Cette connexion sera utilisée pour plusieurs requêtes/réponses.
  - Le client envoie la 1re requête HTTP. Par ex.

```
GET /index.html HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

  - La méthode HTTP peut être :
    - GET
    - HEAD
    - POST
    - etc.
  - Le chemin d'accès de la ressource, *Path*, fait partie de l'URL.

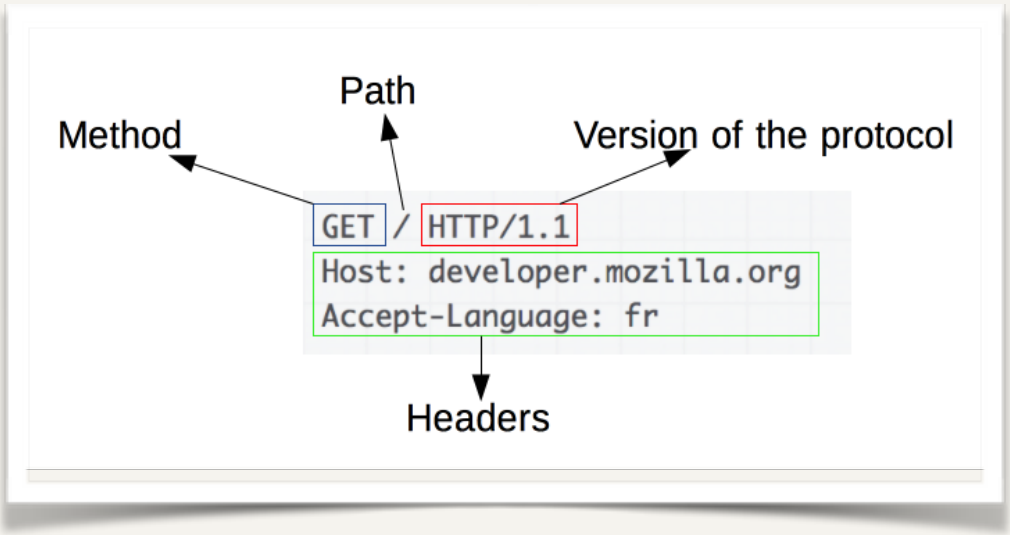


Fig 6.5 - Requête HTTP

Flux HTTP, suite

- Le client lit la réponse du serveur. Par ex.

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

<!DOCTYPE html... (les 29769 octets de la page web
demandée)
```

- La réponse comprend la **version** du protocole, un **code** de status (200, 404, 500…), un **message** de statut (*OK, Not found, Internal server error,…*), les **en-têtes** HTTP, puis un éventuel **corps** avec la ressource demandée.

Les standards du web

- URL, *Uniform Resource Locator*, Format de nommage de ressources ; adresse réticulaire ; (usuellement) adresse web
  - Une URL se décompose en général en quatre principales parties :

```
https://altavista.digital.com:8080/find.pl?q=url&lang=fr
```

Protocole | Nom de domaine | Port | Accès à la ressource

  - Le nom du **protocole** (http ; https ; ftp ; file ; mailto ; news...), suivi par « : »
  - Le nom du serveur : le **nom de domaine** du serveur, précédé par « // »
    - le nom de domaine est composé de sous-domaines optionnels, d'un domaine de 2e niveau et d'un domaine de 1er niveau (TLD, Top Level Domain)

Les standards du web

- **URL**, *Uniform Resource Locator*, suite…

```
https://archive.digital.com:8080/seq/find.php?q=url&lg=fr
        -------  --------  ---       -----  --------- -------------
        Sous-domaine domaine 2e niv.  TLD      Chemin      page      chaine de requête
 ------  ---------------------  ------ ---------------------------
Protocole     Nom de domaine      Port      Accès à la ressource
```

- (en option) Le numéro de **port** TCP ; pour HTTP, le n° de port par défaut est **80** (et 443 pour HTTPS)
- L'**accès à la ressource** ; tous ses éléments sont optionnels :
  - le **chemin** absolu sur le service contenant la page web. Il commence par « / ». Par défaut, « / ».
  - la **page** web.
  - Une chaine de requête, précédée de « ? », se compose en général de paramètres 'clé=valeur' séparés par « & ».
  - Voir : Anatomie d'une URL

Les standards du web

- **HTML**, *HyperText Markup Language*, langage de balisage d'hypertexte
  - Beaucoup de versions depuis 1989. Aujourd'hui : **HTML 5**.
  - C'est le langage de description de page web, constitué de balises et interprété par le navigateur.
  - HTML est lié à la structure d'une page web.
  - Ex. : utc505.seancetenante.com/minimum/page-simple.html
- **CSS**, *Cascading Style Sheets*, Feuille de styles en cascade
  - Langage permettant la mise en page et la présentation du contenu de pages web.
- **JavaScript**
  - JavaScript est un langage de script, favorisant la programmation événementielle et interagissant avec le **DOM**, *Document Object Model*. DOM décrit la structure et le contenu des pages web.
  - JavaScript est utilisé coté client (*client side scripting*), mais parfois aussi coté serveur (*server side scripting*).

Les standards du web

- CGI ; Scripts coté serveur
  - CGI , *Common Gateway Interface* (littéralement : Interface de passerelle commune...)
    - Avec les formulaires (depuis HTML 2.0, en 1995), il est devenu nécessaire de construire coté serveur des pages web **dynamiques**, générées en fonction de données saisies par l'utilisateur.
  - Scripts coté serveurs ; *Server side scripting*
    - Les CGI sont des programmes compilés ou (très souvent) interprétés. Les langages serveur suivants sont populaires :
      - PHP (*PHP HyperText Preprocessor*)
      - Python
      - javascript (avec Node.js)

Sureté de fonctionnement (safety)

- Cela concerne les moyens utilisés pour éviter les dysfonctionnements du système.
- Les risques sont :
  - Panne ou défaillance d'équipements de traitements
  - Perte d'information par dysfonctionnement de mémoire de masse
  - Défaut des équipements réseau
  - Panne de la fourniture d'énergie
  - Incendie et inondation
  - Vols et vandalisme

- Par exemple, pour la sécurisation de système de stockage :
  - On évite les attachements directs de disques-dur aux serveurs (DAS, *Direct Attached System*), pour préférer les systèmes :
  - NAS, *Network Attached Storage* : système de stockage mutualisé sur le réseau local
  - SAN, *Storage Area Network* : Réseau haute disponibilité dédié au stockage.

Sureté de fonctionnement (safety)

- Pour la sécurisation de **l'accès au réseau** :
  - Redondance des liens obtenue par maillage du réseau
  - doublement des raccordements au réseau de l'opérateur
  - ou utilisation d'un lien de secours
  - ou choix de multiples opérateurs :
    - avec équilibrage de charge en fonctionnement normal
    - en cas de rupture d'un lien, utilisation du lien restant (fonctionnement en mode dégradé).
- Contre les défaillances du **réseau électrique** :
  - Un **onduleur off-line** protège un poste de travail (ou un serveur)
  - Un **onduleur on-line** alimente en permanence un réseau de distribution électrique local dédié aux moyens informatiques
  - On tient compte d'une autonomie d'environ 20 mn pour ces onduleurs
  - Si nécessaire, un **groupe électrogène** va se substituer au fournisseur d'énergie défaillant.

Introduction à la sécurité

1 - Sureté et sécurité

Sécurité (security)

- Cela regroupe les moyens et mesures prises pour mettre le système à l'abri de toute agression.

- La sécurité informatique vise cinq grands objectifs :
  - ① L'**Intégrité** : garantir que les données sont bien celles que l'on croit être.
  - ② **Disponibilité** : maintenir le bon fonctionnement du système d'information et garantir d'accès aux services et aux ressources.
  - ③ **Authentification** : assurer l'identité d'un utilisateur.
  - ④ La **confidentialité** : assurer que seules les personnes autorisées aient accès aux ressources échangées (cela nécessite authentification et d'autorisation d'accès).
  - ⑤ **Non-répudiation** : garantir qu'aucun partenaire ne pourra nier une transaction.

Sécurité (security)

- Ceci est à mettre en opposition respectivement aux **menaces, risques ou attaques** suivants :
  - ① Altération de données ; injection de données.
  - ② Pannes matérielles ou logicielles ; congestion ; déni de service.
  - ③ Usurpation d'identité.
  - ④ prendre connaissance de données sans y être habilité.
  - ⑤ désaveu.
- Comment ?
  - ① Calcul d'une somme de contrôle ou d'un résumé de message (fonction de hashage).
  - ② Système à tolérance de pannes ; redondance.
  - ③ Système d'authentification (Échange de mot de passe, utilisation d'une clé, etc.) et droit d'accès (ACL, *Access Control List*).
  - ④ Chiffrement de données (cryptographie) ; contrôle d'accès.
  - ⑤ Utilisation d'un tiers de confiance (Notaire) qui enregistre et authentifie les transactions.

Origine des risques

- Physique :
  - Désastre naturel (inondation, séisme, incendie)
  - Intempéries
  - Panne matérielle
  - Panne du réseau
  - Coupure électrique

- Origine humaine :
  - Erreur de conception (du logiciel, mauvais dimensionnement...)

- Attaque avec accès physique :
  - Coupure de l'électricité
  - Extinction manuelle d'ordinateur ou d'équipement
  - Vandalisme
  - Vol (ordinateurs ou disques durs)
  - Écoute du trafic sur le réseau

Origine des risques

- Interception de communications
  - Vol de session (session hijacking)
  - Usurpation d'identité (Ex. : IP spoofing)
  - Détournement ou altération de messages
- Dénis de service
  - Inondation
  - Perturbation des connexions
- Intrusion
  - Balayage de ports
  - Élévation de privilèges (par débordement de tampon par ex.)
  - Chevaux de Troie, vers et virus
  - Trappe (*backdoor*) ; porte dérobée dans un logiciel
- Ingénierie sociale
  - *Phishing*, hameçonnage : escroquerie par e-mail, par téléphone ou site web falsifié en vue d'une arnaque ou pour obtenir des renseignements personnels ou des mots de passe.

Politique de sécurité

- C'est un document de référence dans une entreprise qui fixe des règles, procédures et bonnes pratiques pour assurer un niveau de sécurité conforme aux besoins.
  - Sensibilisation des utilisateurs à la sécurité informatique
  - Organisation de la sécurité : SOC, *Security Operations center* : équipe en charge d'assurer la sécurité de l'information
  - Surveillance des événements de sécurité (SEM)

Méthodes

- MÉHARI (Méthode harmonisée d'analyse des risques) du CLUSIF (Club de la Sécurité de l'Information Français)
- EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) de l'Agence nationale de la sécurité des systèmes d'information (ANSSI).
- la méthode OCTAVE (*Operationally Critical Threat, Asset, and Vulnerability Evaluation*), développée par l'Université de Carnegie Mellon (USA)
- Gestion des risques informatiques : Normes ISO 27000 :
  - Systèmes de management de la sécurité de l'information Vue d'ensemble et vocabulaire.

Exemples d'attaques

- **Fraude au président (Pathé)**
  - Le groupe de cinéma Pathé a été victime d'une attaque de phishing avancée de type "*Whaling*" entrainant une escroquerie de près de 19 millions d'euros.
  - Des pirates se sont fait passer pour des dirigeants de l'entreprise auprès de sa filiale localisée aux Pays-Bas.
  - Pathé a été la cible d'un groupe de fraudeurs professionnels qui, grâce à une communication raffinée, a réussi à gagner la confiance de certains collaborateurs.
  - Plusieurs collaborateurs ont ainsi été contactés afin de procéder à des virements d'argent importants pour un total de 19,2 millions d'euros.
  - Des emails, envoyés au directeur financier de Pathé aux Pays-Bas et à la directrice de Pathé Pays-Bas exigeaient ainsi des transferts d'argent dans le but de mener une importante acquisition à Dubaï.

Exemples d'attaques

- **Cyberattaque NotPetya**
  - Troisième cyberattaque en moins de deux mois après *WannaCry* et *Adylkuzz*, en juin 2017.
  - C'est un logiciel malveillant de type *wiper* (il détruit les données), mais apparait sous la forme d'un rançongiciel (*ransomware*).
  - Il touche toutes les versions de Microsoft Windows, de Windows XP à Windows 10
    - Comme *WannaCry,* il utilise pour se propager la faille de sécurité *EternalBlue* volée à la NSA par un groupe de pirates informatiques *The Shadow Brokers*.
    - Cette faille de sécurité de Windows a été corrigée par Microsoft en mars 2017, mais beaucoup d'entreprises n'ont pas mis à jour leur système d'exploitation, d'où cette cyberattaque mondiale.
  - Sur un écran noir avec un texte écrit en rouge et en anglais, le message suivant s'affiche à chaque démarrage de l'ordinateur :
    - *Ooops, vos fichiers ont été chiffrés. Si vous voyez ce message, vos fichiers ne sont plus accessibles, car ils ont été chiffrés. Peut-être que vous recherchez un moyen de récupérer vos fichiers, mais ne perdez pas votre temps. Personne ne peut récupérer vos fichiers sans notre service de déchiffrement.*

Adopter une politique de mot de passe rigoureuse

- Un mot de passe doit contenir au moins 12 caractères, incluant des majuscules, des minuscules, des chiffres, et des caractères spéciaux.
- Éviter les mots communs et des informations personnelles (la date de naissance, etc.)
- Utiliser des mots de passe uniques pour chaque compte.
- Renouveler les mots de passe tous les 90 jours ou immédiatement en cas de compromission
- Ne jamais partager ses mots de passe.
- Utiliser un gestionnaire de mots de passe pour stocker et générer des mots sécurisés.
- Activer l'authentification multifacteur (MFA, *Multi-Factor Authentication*).
- Réaliser des audits réguliers pour vérifier la sécurité des mots de passe.
- Former le personnel sur les bonnes pratiques et les risques liés à la gestion des mots de passe.

Introduction à la sécurité

2 - Bonnes pratiques de sécurité personnelle

Sauvegarder ses données régulièrement

- Utiliser un service de stockage en ligne (pCloud, kDrive, Google Drive, etc.)
- Sauvegardes sur un disque dur externe.
- Sauvegardes sur un NAS, *Network Attached Storage*.

Faire ses mises à jour régulièrement

- Identifiez l'ensemble de vos appareils et logiciels utilisés.
- Lorsque l'on vous propose une mise à jour, faites-la immédiatement, ou activez l'option d'installation automatique des mises à jour si elle existe.
- Téléchargez les mises à jour uniquement depuis les sites officiels des éditeurs.
  - Méfiez-vous de fausses mises à jour que l'on vous propose sur Internet. Vérifiez toujours l'URL du site sur lequel vous vous trouvez.

Introduction à la sécurité

2 - Bonnes pratiques de sécurité personnelle

Se protéger des virus et autres logiciels malveillants

- Virus, vers, cheval de Troie, ou logiciels espions (spyware) sont des techniques couramment utilisées par les pirates informatiques.
- Pour vous protéger de ces intrusions, il est indispensable de posséder ces deux outils :
  - Un antivirus
  - Un pare-feu bien configuré qui bloquera les connexions non désirées depuis votre ordinateur

Évitez les réseaux Wifi publics ou inconnus

- Désactivez les connexions sans-fil (Wifi, Bluetooth, NFC, …) lorsque vous ne vous en servez pas.
- Privilégiez la connexion privée 3G ou 4G associée à votre abonnement mobile.
  - Sécuriser le partage de connexion de vos appareils à l'aide d'un mot de passe.
- Si vous devez utiliser un Wifi public :
  - Ne jamais y réaliser d'opérations à caractère sensible (paiement par carte bancaire, etc.)
  - Si possible utilisez un réseau privé virtuel (VPN).

Bien séparer ses usages professionnels et personnels

- Équipements séparés :
  - Utilisez des appareils différents pour vos activités professionnelles et personnelles.
  - À défaut, créez des comptes utilisateurs distincts pour chaque usage afin de cloisonner les données.
- Stockage sécurisé : Ne stockez pas de données professionnelles sur des équipements ou services de stockage en ligne personnels.
- Messageries distinctes : Ne mélangez pas vos emails professionnels et personnels.
  - Utilisez des applications de messagerie dédiées pour chaque type de communication afin d'éviter les erreurs de destinataire et les fuites de données

Éviter de naviguer sur des sites douteux ou illicites

- Être vigilant lors du téléchargement d’un fichier.

Être vigilant sur les liens ou les pièces jointes contenus dans des messages électroniques

- L'hameçonnage (ou phishing) désigne une technique frauduleuse qui consiste à usurper l'identité d'un organisme connu (banque, opérateurs, etc) ou d'un proche pour récupérer des informations confidentielles.
  - Ne communiquez pas d'informations personnelles ou professionnelles par messagerie ou par téléphone.
  - En cas de réception d'un message contenant un lien, positionnez le pointeur de la souris (sans cliquer) sur ce lien pour afficher l'adresse vers laquelle il pointe réellement.
  - Vérifiez bien l'adresse du site internet avant de renseigner des données. Si cela ne correspond pas exactement au site concerné, il s'agit certainement d'un site frauduleux.
  - Activez si possible la double authentification pour sécuriser vos accès.
  - Utilisez des mots de passe de différents et complexes pour chaque site et application.
  - Saisissez directement dans votre navigateur l'adresse du site concerné.

Notions de cryptographie

- Il s'agit d'appliquer une **méthode de chiffrement** à l'aide d'une **clé de chiffrement** à un **message clair** pour obtenir un **cryptogramme** (message crypté).
- Seul le **cryptogramme** est transmis sur le réseau.
- Le destinataire effectue le déchiffrage avec une **méthode de déchiffrement** grâce à une **clé de déchiffrement**.

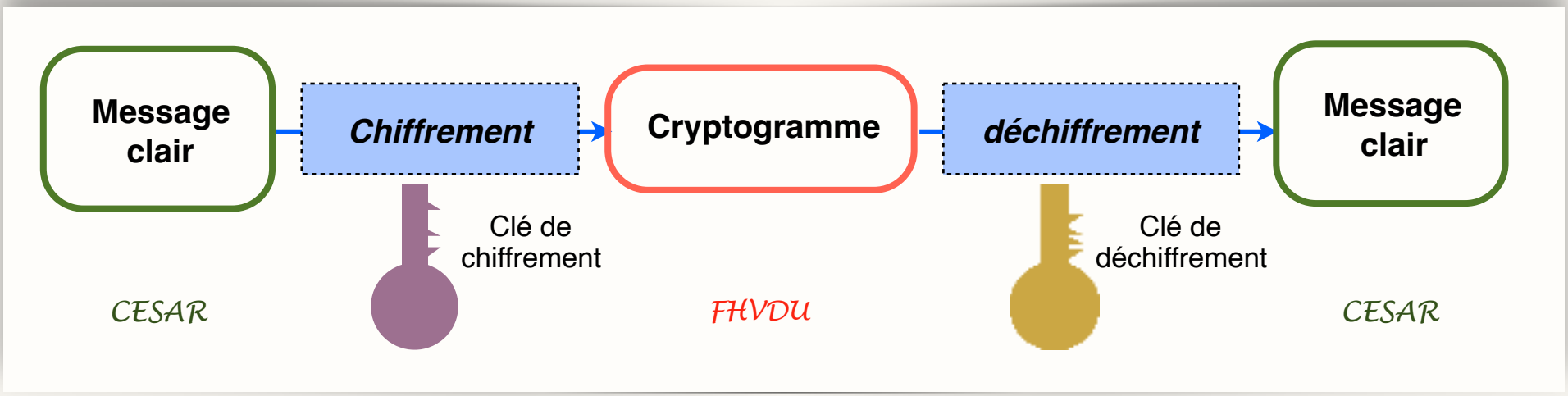


Fig 7.1 - Méthodes de chiffrement et déchiffrement

Cryptographie ; principe de Kerckhoffs

- La sécurité d'un cryptosystème ne doit reposer que sur **le secret de la clé.**
- Tous les autres paramètres doivent être supposés publiquement connus
  - Méthode de chiffrement
  - Méthode de déchiffrement
- Quand les systèmes de chiffrement sont publics, largement étudiés et qu'aucune attaque significative n'est connue, ils sont d'autant plus sûrs.

Chiffrement symétrique

- La même clé (ou certificat) est utilisée pour le chiffrement et le déchiffrement.
- Méthodes rapides et économes en ressource.
- Ex. : Code de César, CAST, DES (*Data Encryption Standard*), 3DES, AES (*Advanced Encryption System*)...
- Assure des fonctions de **confidentialité**, **authentification** et **intégrité**.
- Problème : **partage et transmission de la clé**.

Chiffrement asymétrique

- Usage d'une **paire de clés** asymétrique :
  - la clé **publique** (*public key*) est utilisée pour crypter les messages adressés au propriétaire de la clé privée.
  - la clé **privée** (clé secrète, *private key*) est utilisée pour le déchiffrement de cryptogrammes chiffrées à l'aide de la clé publique.

- Algorithmes gourmands en ressource.
- Ex. : RSA (Rivest, Shamir et Adleman), Rabin, ElGamal, DSA (*Digital Signature Algorithm*)...
- Applications : **confidentialité**, **authentification**, **intégrité** et **partage de clé** de chiffrement symétrique.

Chiffrement asymétrique

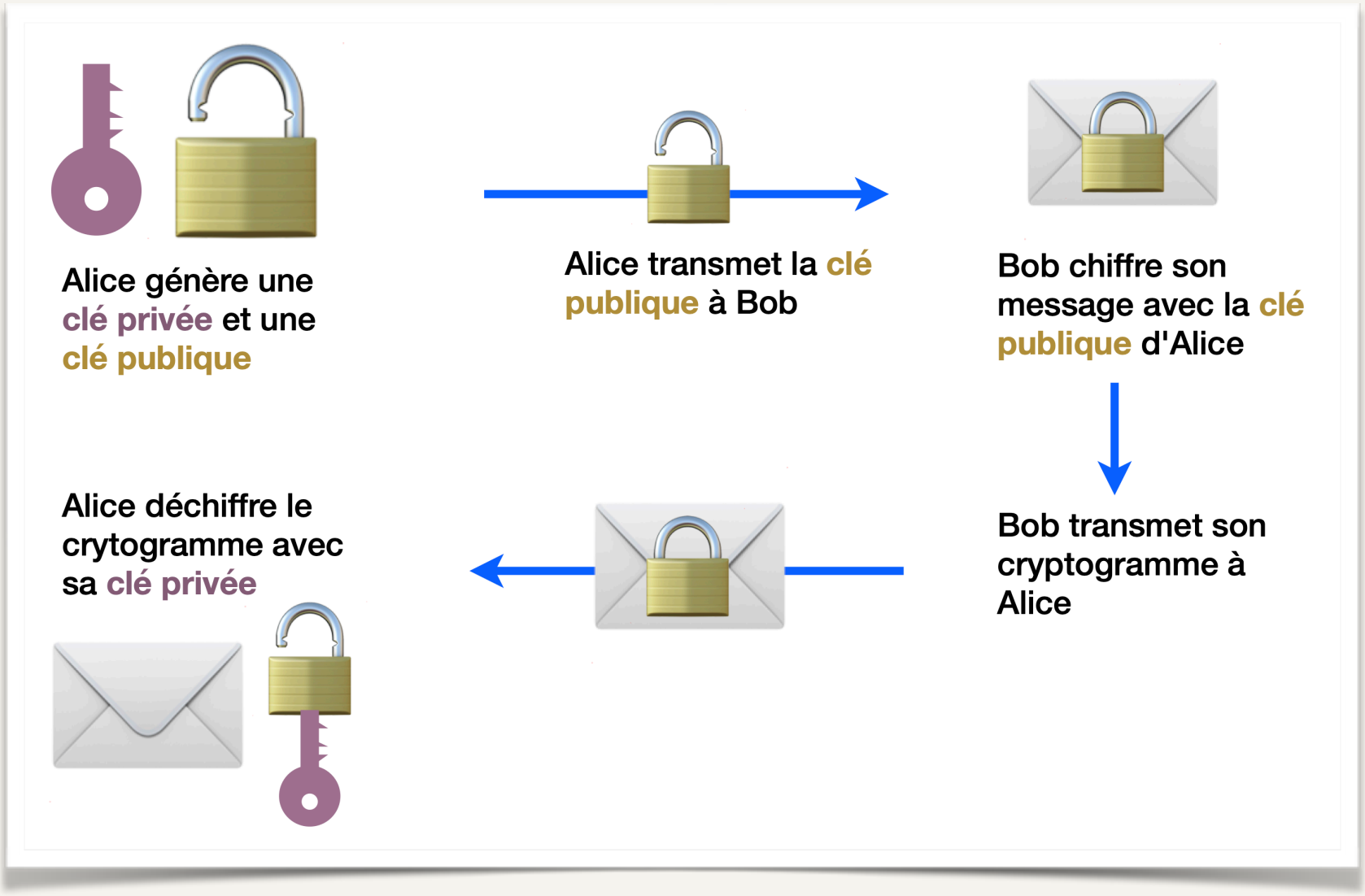


Fig 7.3 - Chiffrement asymétrique

Exemple d'une authentification de l'émetteur

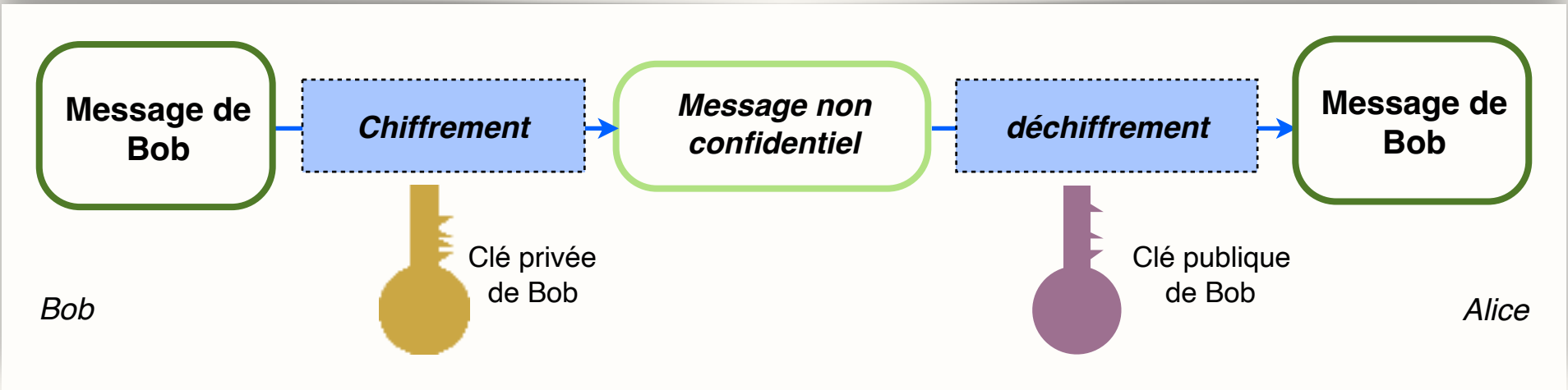


Fig 7.3 - Méthode d'authentification

- Bob chiffre son message avec sa clé privé.
- Alice déchiffre le message avec la clé publique de Bob. Cela prouve que ce message a été codé avec la clé privée de Bob et cela **authentifie** Bob comme émetteur du message.
- C'est un système d'**authentification**, mais pas de confidentialité (car tout le monde peut déchiffrer le message de Bob).

Fonction de hachage

- Elle a trois propriétés :
  - Le résultat (résumé de message ou hash ou digest ou empreinte) est toujours de la même taille. 128 bits pour MD5.
  - La fonction de hachage est à sens unique : on ne peut pas retrouver le message de départ à partir du digest.
  - Deux messages différents (même très légèrement) ont des digests très différents.
- Exemple :
  - MD5 (Message digest) : voir utc505.seancetenante.com/securite/md5.php
  - SHA-1 (Secure Hash Algorithm)
  - SHA-256 : voir utc505.seancetenante.com/securite/sha256.php
- Application :
  - Intégrité de données
  - Signature (digest chiffré avec une clé privé => sceau)

Filtrage par le routeur d'accès

- Un routeur d'accès peut assurer un filtrage simple par analyse des adresses IP sources et destination
- Il agit au niveau 3 (couche réseau), à l'aide de listes d'adresses acceptées ou refusées.
- ACL = **Access Control List**

La traduction d'adresse

- NAT = Network Address Translation
- NAPT = Network Address and Port Translation
- C'est un moyen de masquer le plan d'adressage de l'entreprise
- (et de pallier à la pénurie d'adresses IPv4...)
- NAPT permet notamment de faire correspondre une seule adresse externe publique visible sur internet à toutes les adresses d'un réseau privé.

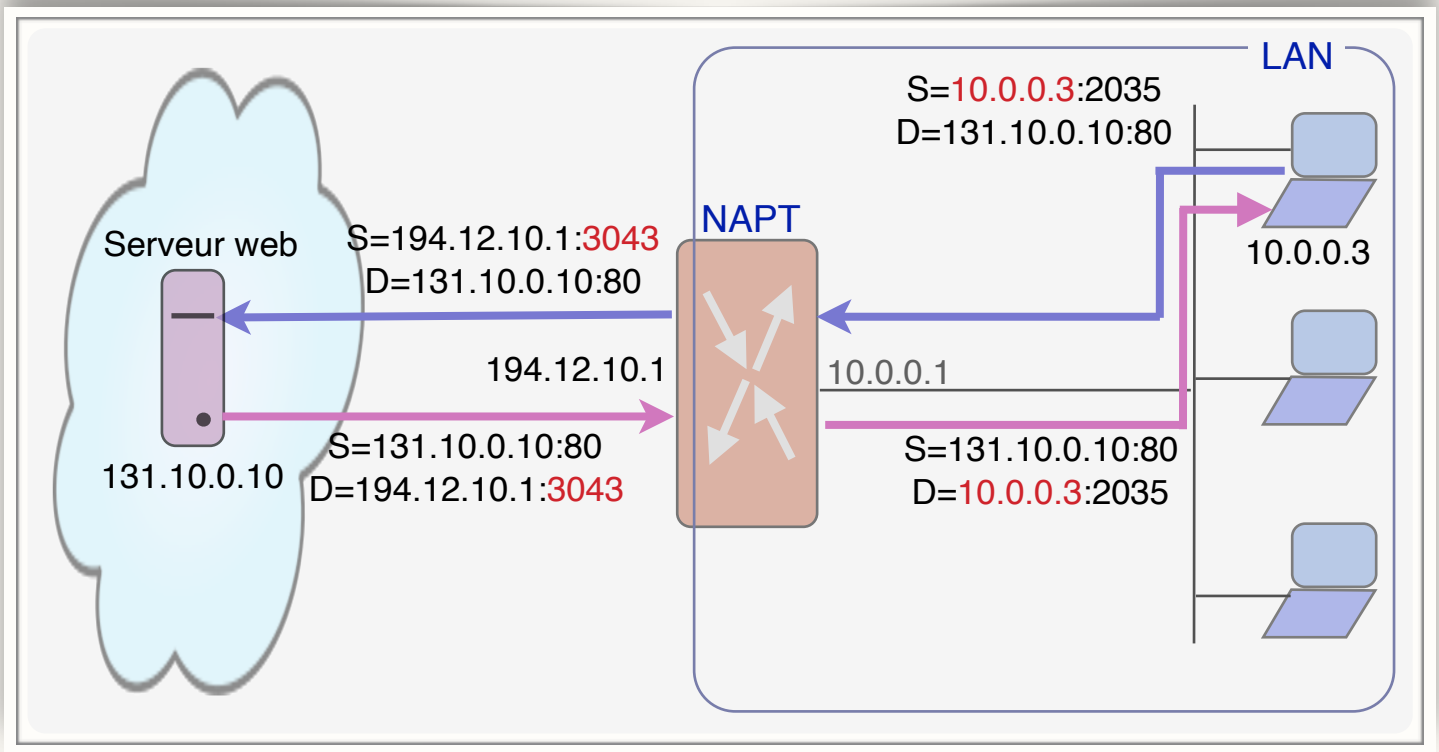


Fig 7.4 - Principe du NAPT

Pare-feu ; Firewall

- Un pare-feu (ou coupe-feu) offre un filtrage évolué pour faire respecter la politique de sécurité du réseau.
- Chaque paquet reçu est examiné ; il est rejeté ou accepté en fonction de :
  - adresse destination
  - port destination
  - adresse source
  - port source
  - protocole transporté (ICMP, UDP, TCP...)
  - la valeur de certains drapeaux d'en-tête
  - etc.
- Un pare-feu peut être configuré en tenant compte d'une DMZ : ***DeMilitarized Zone***, soit zone de sécurité.
  - Par ex. dans la DMZ, un serveur web n'aura pas les filtrages définis pour le reste du réseau.

Pare-feu ; Firewall

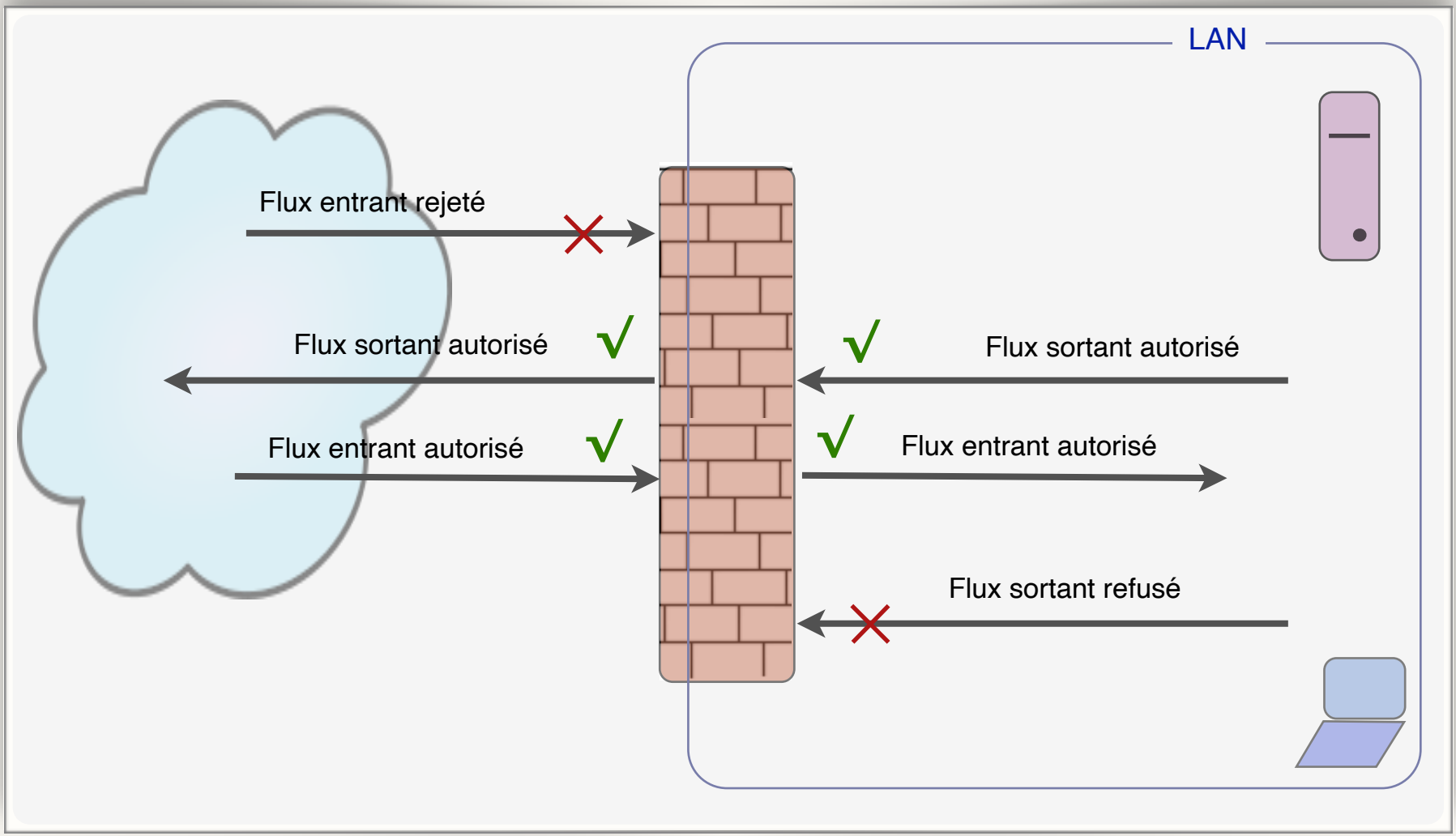


Fig 7.5 - Principe du pare-feu

Pare-feu ; Firewall

- Différentes catégories sont à considérer :

- **Pare-feu sans état** : il intégrait d'anciens routeurs, afin d'examiner les paquets indépendamment les uns des autres, suivant des règles nommées, suivant les constructeurs :
  - ACL : Access Control List (Cisco)
  - Policy (Juniper)
  - Règles, filtres, etc.

- **Pare-feu à états** (stateful firewall). Suivant le type de transport :
  - TCP : il vérifie que les paquets sont liées à une même connexion TCP et sont conformes aux ACL.
  - UDP : si les ACL autorise un datagramme UDP du fait du quadruplet (ip_src, port_src, ip_dest, port_dest) la réponse (avec le quadruplet inverse) sera également acceptée.

Pare-feu ; Firewall

- **Pare-feu applicatif** (Application Layer Gateway ou Proxy-Server) : des ensembles de paquets sont décapsulés pour examiner la conformité au niveau applicatif.
- Un tel pare-feu est composé de deux routeurs, filtrants des paquets (au niveau 3) et d'une passerelle d'application qui permet ce filtre plus conséquent
- Le filtrage est donc effectué au niveau de chaque service offert
- Des conversions de protocoles sont alors possibles
- Certains virus et chevaux de Troie restent malgré tout indécelables.

VPN (*Virtual Private Network* - Réseau privé virtuel)

- Un VPN permet une extension des réseaux locaux tout en préservant la sécurité logique. Il permet :
  - soit une interconnexion de réseaux locaux via une technique de « tunnel » (tunneling)
  - soit un moyen d'accès au système d'information pour des utilisateurs nomades.

- Internet est souvent utilisé comme support de transmission en utilisant un protocole de « tunnellisation », c'est-à-dire encapsulant les données à transmettre de façon chiffrée.

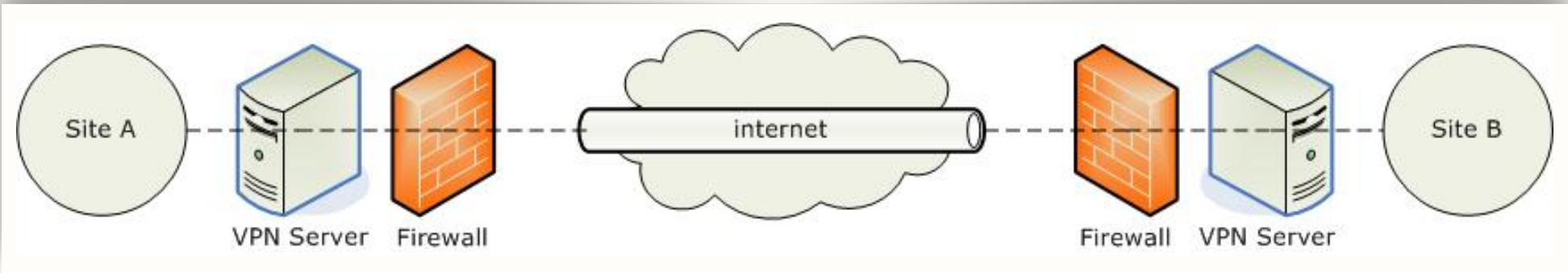


Fig 7.6 - Virtual Private Network

VPN (*Virtual Private Network* - Réseau privé virtuel)

- Le VPN permet donc d'obtenir une liaison sécurisée à moindre coût, si ce n'est la mise en œuvre des équipements terminaux. En contrepartie, il ne permet pas d'assurer une qualité de service comparable à une ligne louée dans la mesure où le réseau physique est public, donc non garanti.

- Le VPN implique :
  - L'authentification (et donc l'identification) du client et du serveur
  - La confidentialité des données par chiffrement

VPN (*Virtual Private Network* - Réseau privé virtuel)

- On distingue les protocoles suivant :
  - IPSec : *Internet Protocol Security Standard*, protocole de niveau 3, offre les services de contrôle d'accès, d'authentification, d'intégrité et de confidentialité de données. Il utilise un mécanisme anti-rejeu et admet un bon nombre d'algorithmes de chiffrement et de hachage.
  - PPTP, *Point-to-Point tunneling Protocol*, protocole de niveau 2 conçu par Microsoft.
  - L2F, *Layer Two Forwarding,* de Cisco. Obsolète
  - L2TP, *Layer Two Tunneling Protocol,* est l'aboutissement des travaux de l'IETF (RFC 3931) pour faire converger les fonctionnalités de PPTP et L2F. C'est un protocole de niveau 2 s'appuyant sur PPP.