



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

UE CNAM - UTC505

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

Document provisoire.

Copie et diffusion non autorisées sans accord écrit.

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



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.
- ❖ 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.
- ❖ Quelques protocoles de la couche application
 - HTTP, DNS...
- ❖ 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.

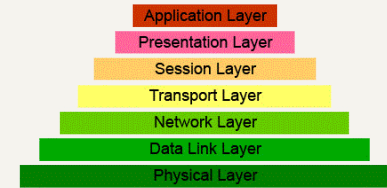


2 - Prologue

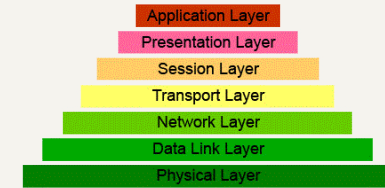
Taille de réseaux

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



- ❖ **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



❖ Illustration

- L'architecture philosophe/
traducteur/
secrétaire

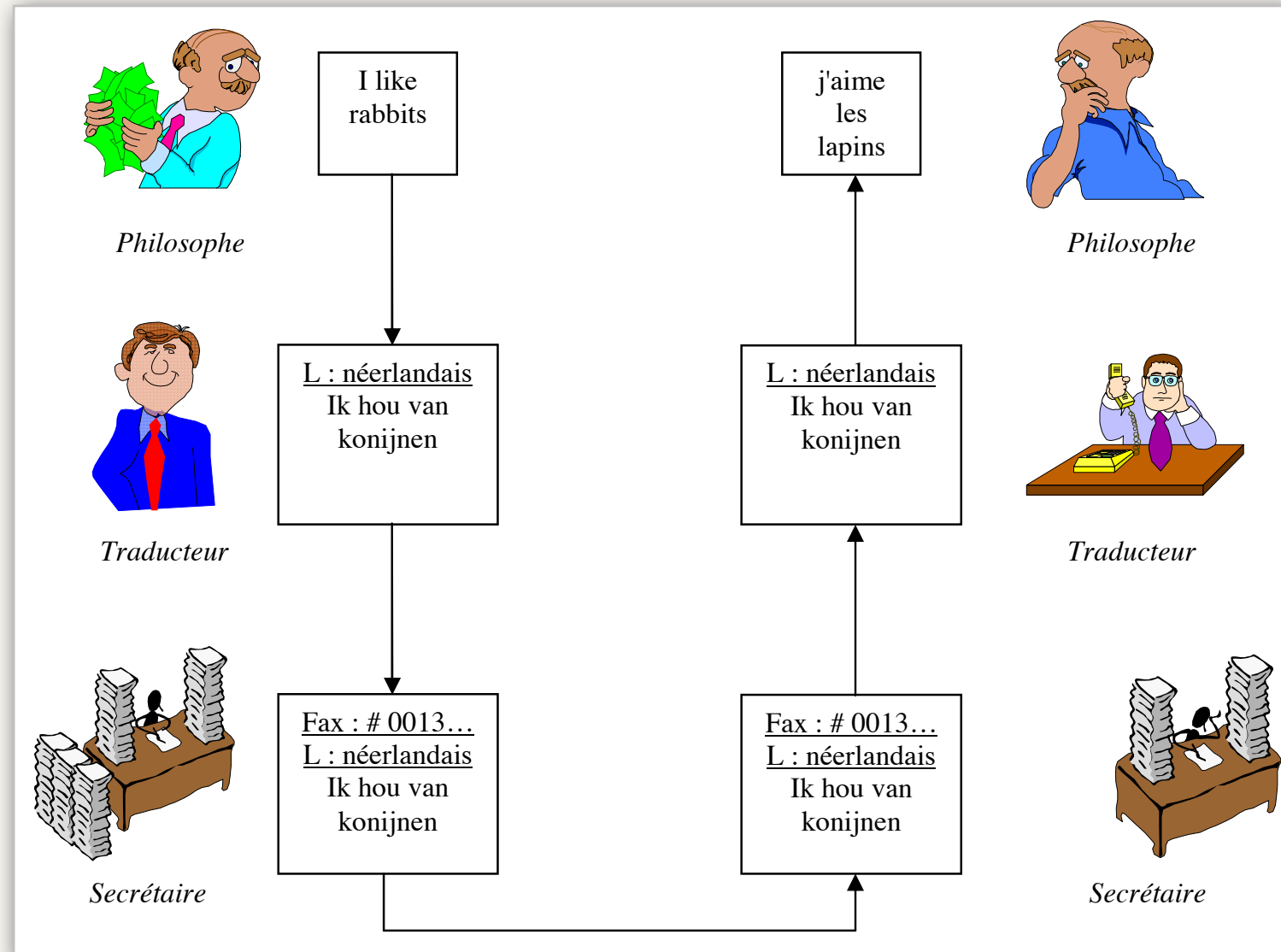


Fig 1.1 - Une architecture à trois couches

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
- ❖ **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 $n/n+1$

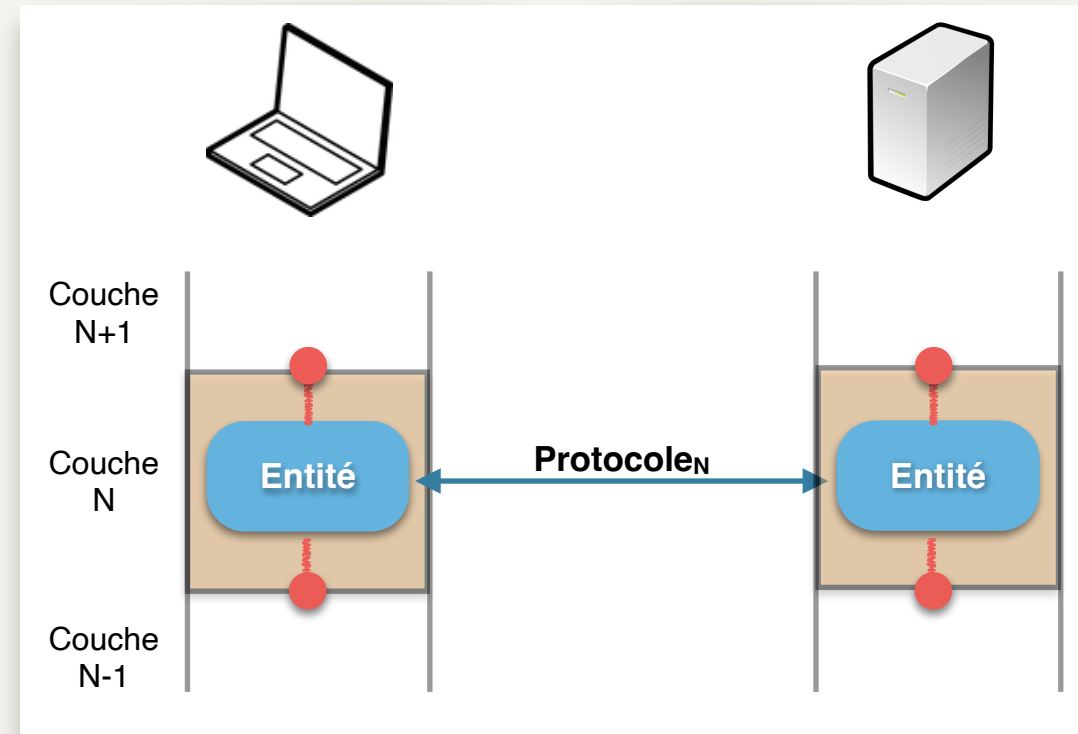
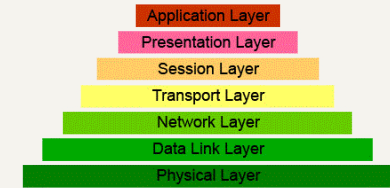


Fig 1.2 - Protocole_N d'une couche N



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

❖ 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.

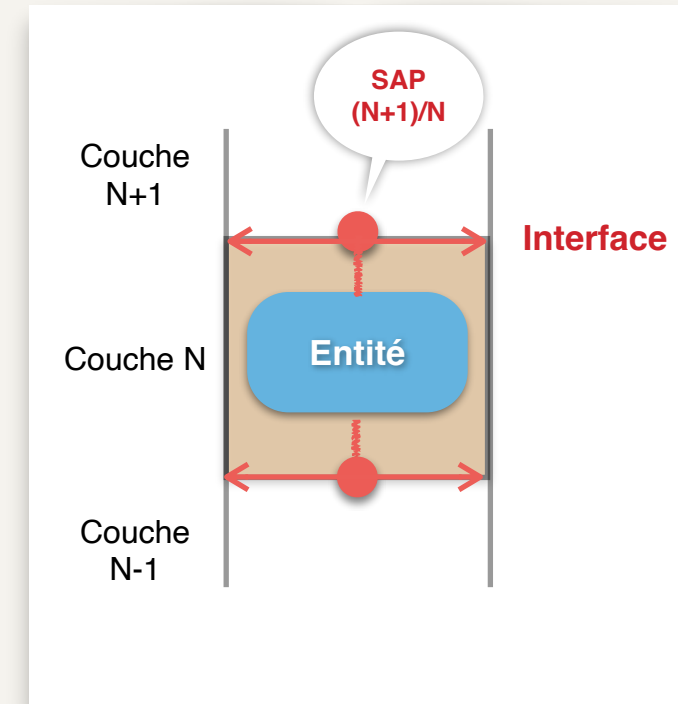


Fig 1.3 - Interfaces entre couches

❖ Décapsulation

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

2 - Définitions

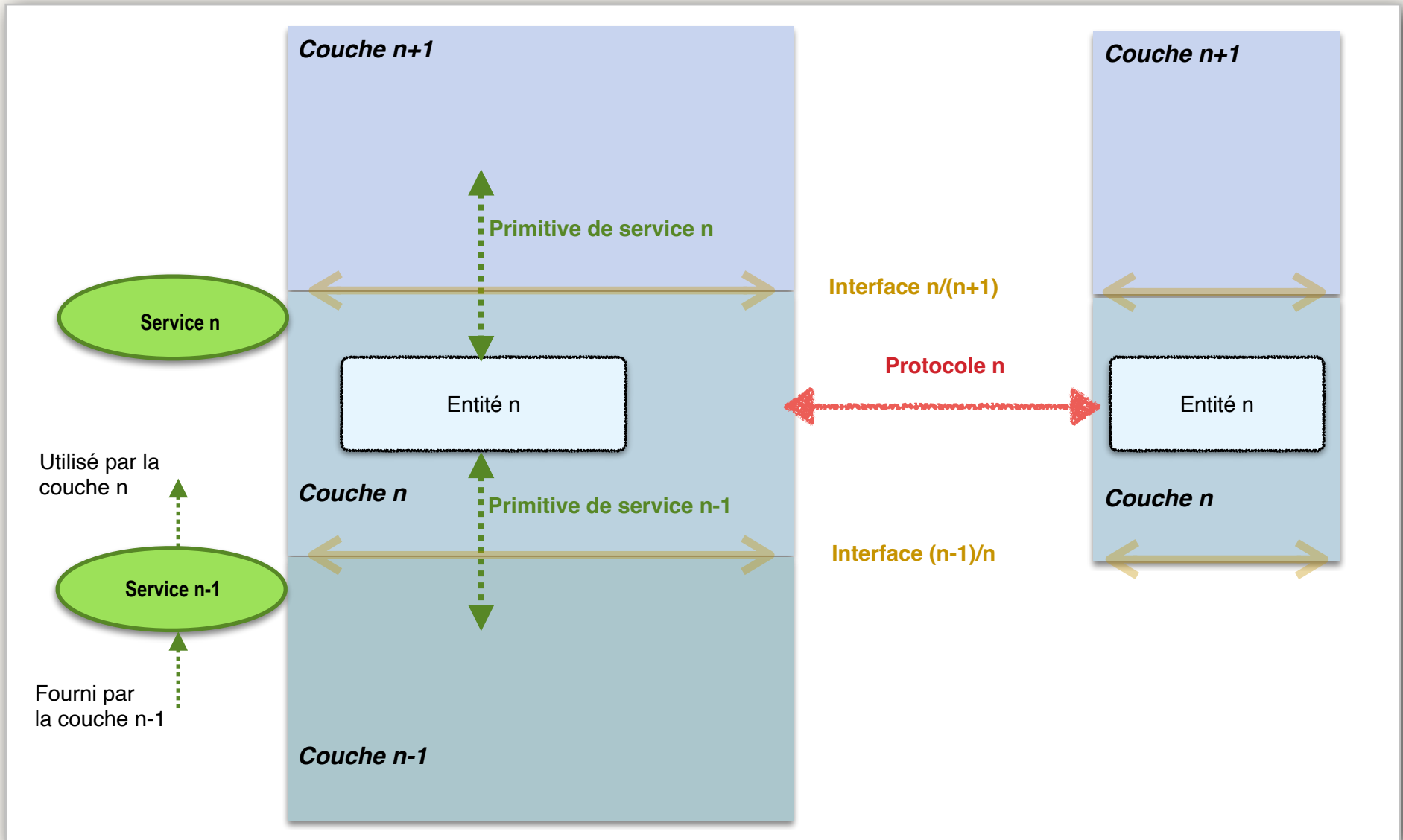
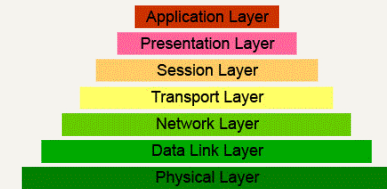


Fig 1.4 - Couches, services et protocoles

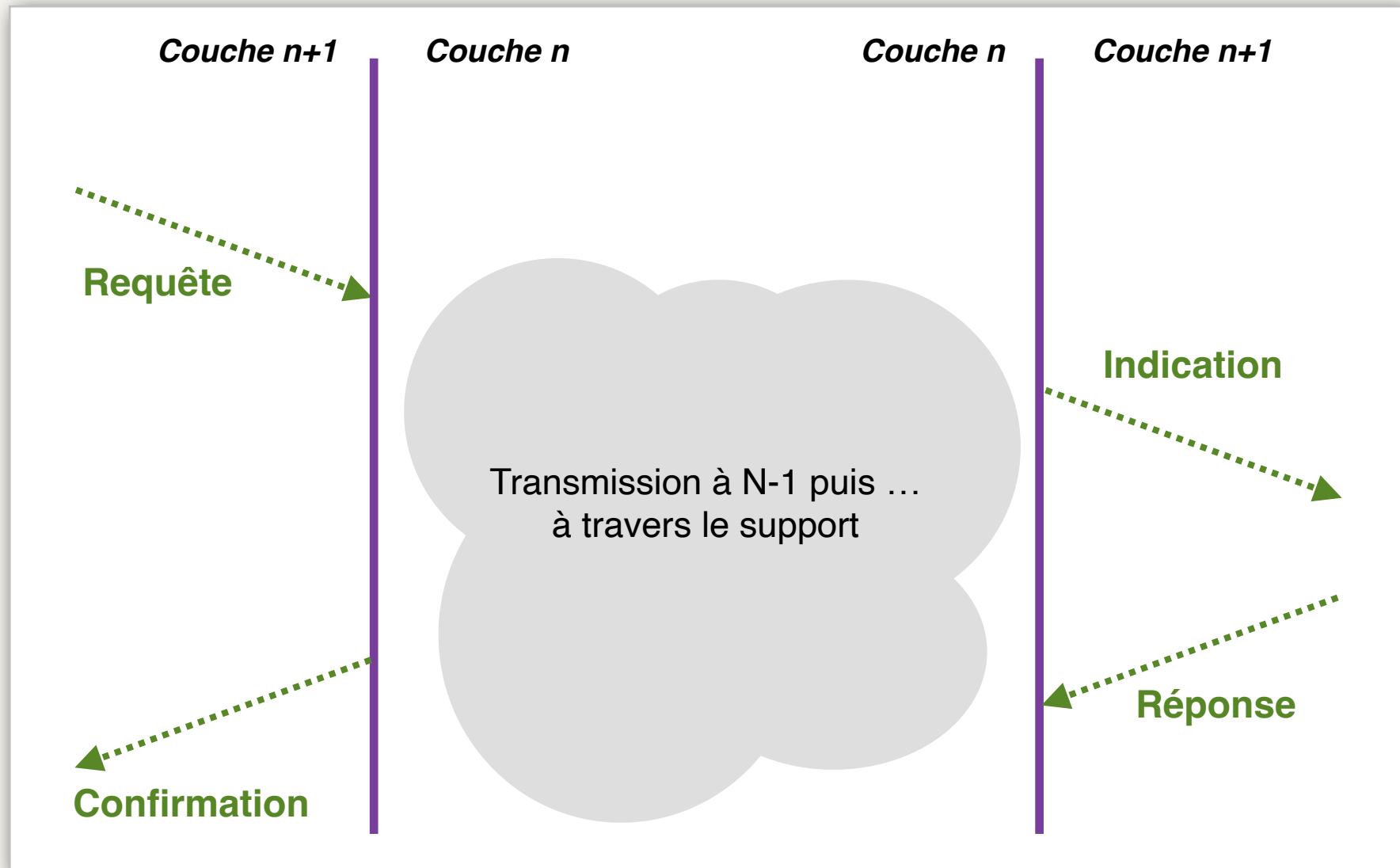
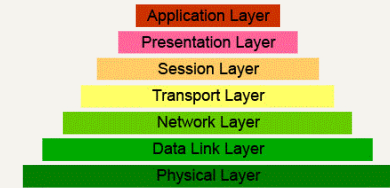


Fig 1.5 - Primitives de service

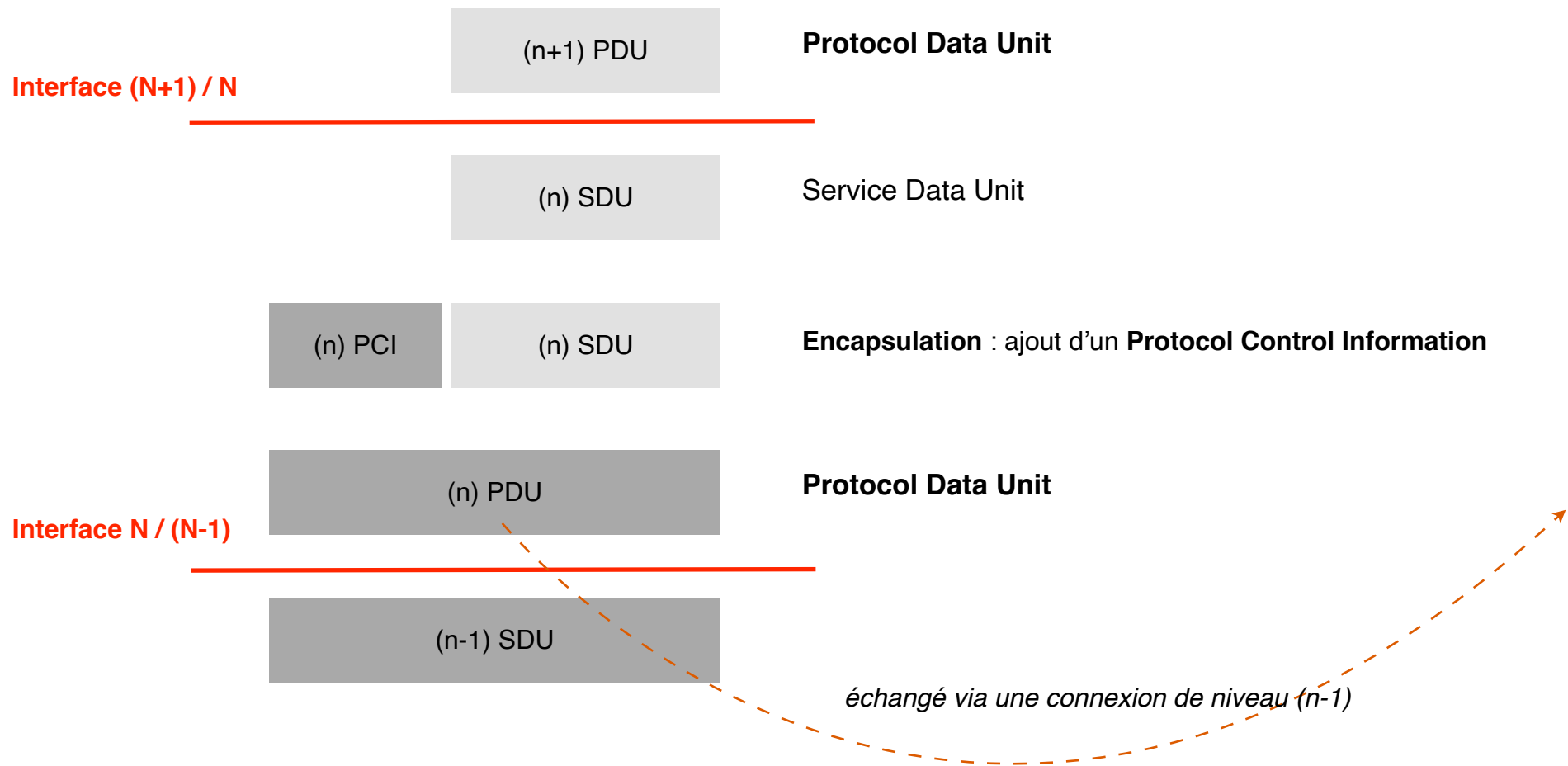
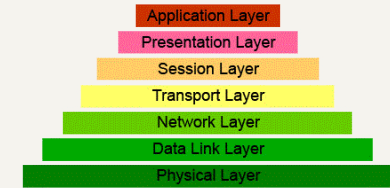


Fig 1.6 - Unités de Données de Protocole & encapsulation

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.

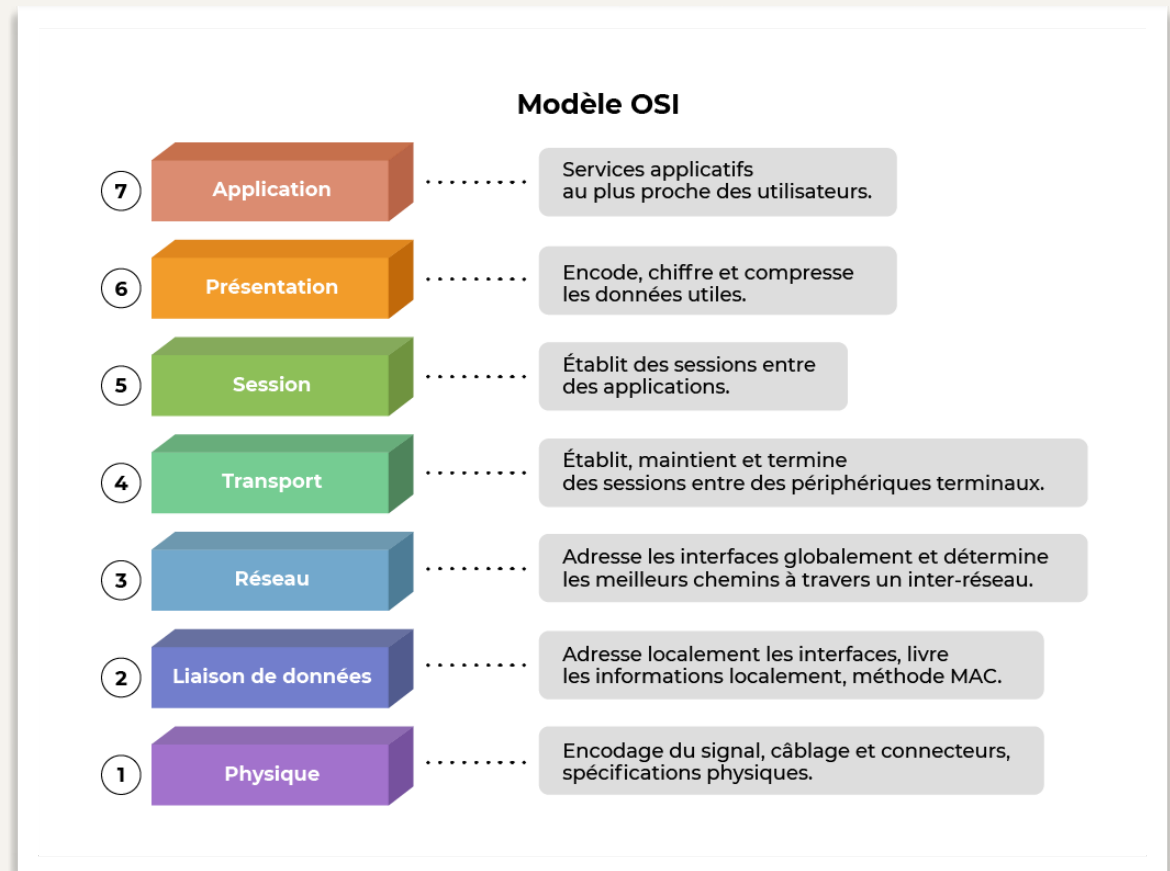
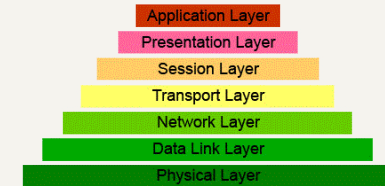


Fig 1.7 - Le modèle OSI

3 - Le modèle de référence OSI



► **Pour**
Le
Réseau
Tout
Se
Passe
Automatiquement

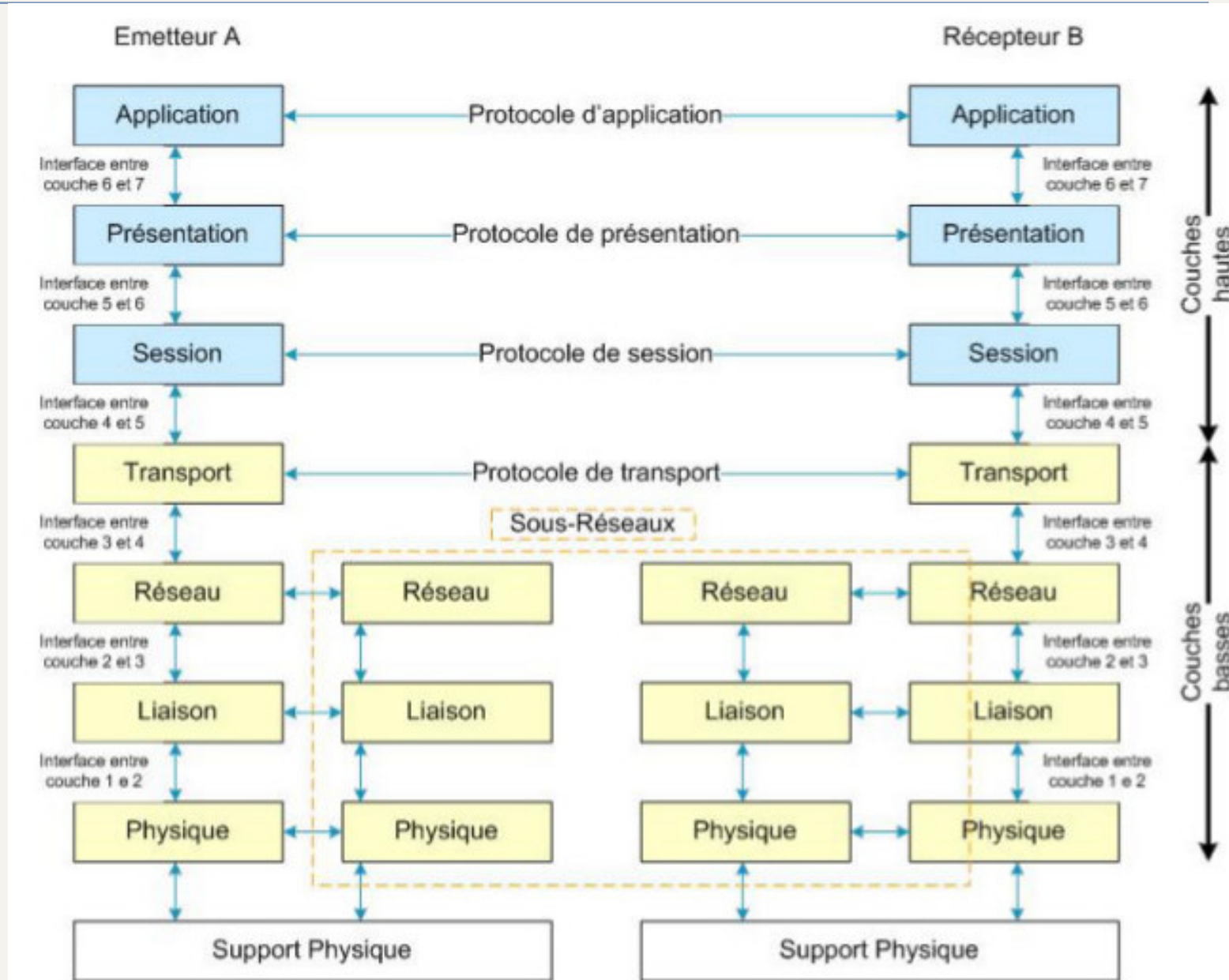
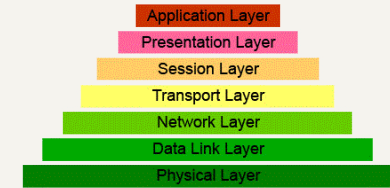


Fig 1.8 - Schéma du modèle en couche OSI

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.

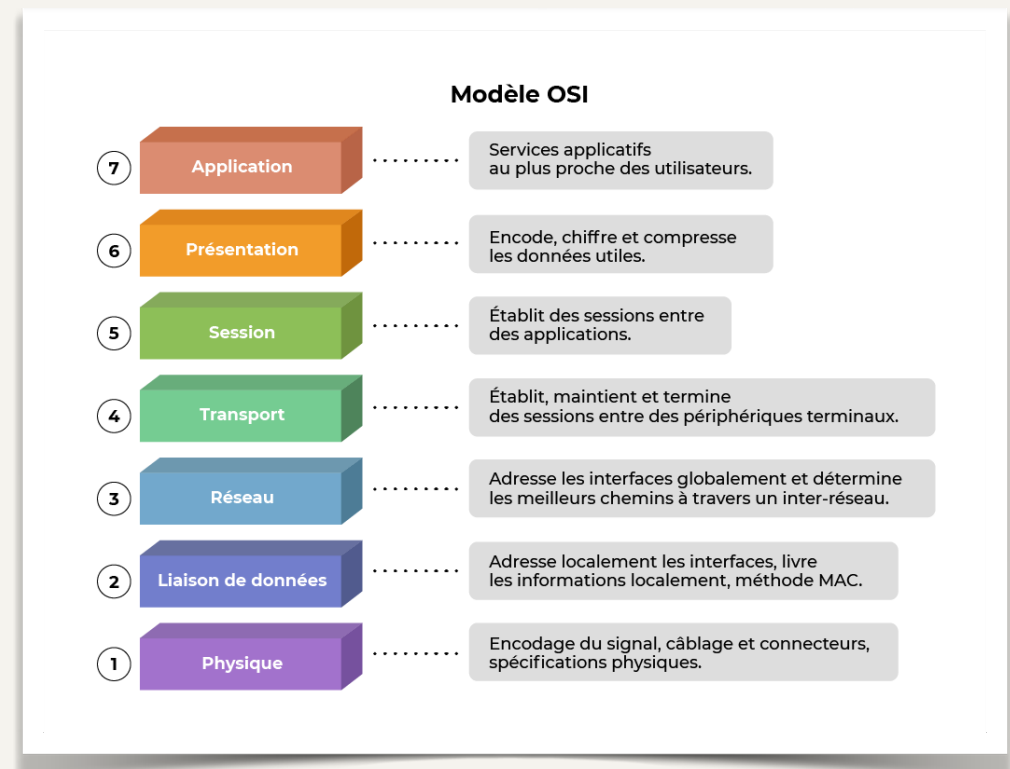
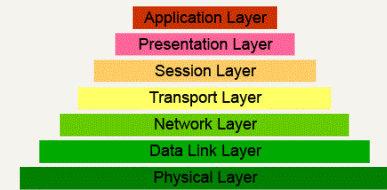


Fig 1.9 - Le modèle OSI

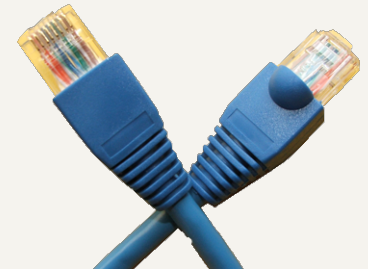


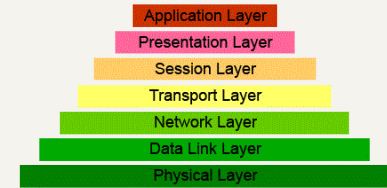
3 - Le modèle de référence OSI

Les sept couches du modèle OSI

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

- Elle concerne le transfert de **train de bits** d'information sur le support physique
- Unité de PDU : **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





3 - Le modèle de référence OSI

Les sept couches du modèle OSI

Couche physique (suite)

- La transformation de trains de bits en **signaux** adaptés au support passe par des techniques de **codage en bande de base** ou de **modulation**.
- **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

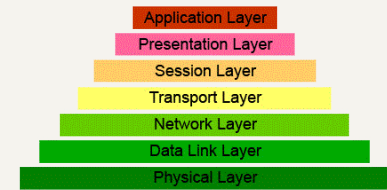
$$D = \frac{V}{T}$$

D : débit binaire en bit/s

V : Volume d'information en bit

T : durée d'émission

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



3 - Le modèle de référence OSI

Les sept couches du modèle OSI

Couche physique (suite)

- **Latence** (*lag*) : Temps de transit entre l'émission d'un bit et sa réception
- **Gigue** (*jitter*) : **Variation** de la latence dans le temps. Une gigue faible indique une connexion stable
- **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)

3 - Le modèle de référence OSI

Les sept couches du modèle OSI

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

- Transfert de **frames** entre deux systèmes adjacents
- Établissement, maintient, contrôle et libération du **lien logique** entre les deux entités
- Constitution et transmission de frames **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

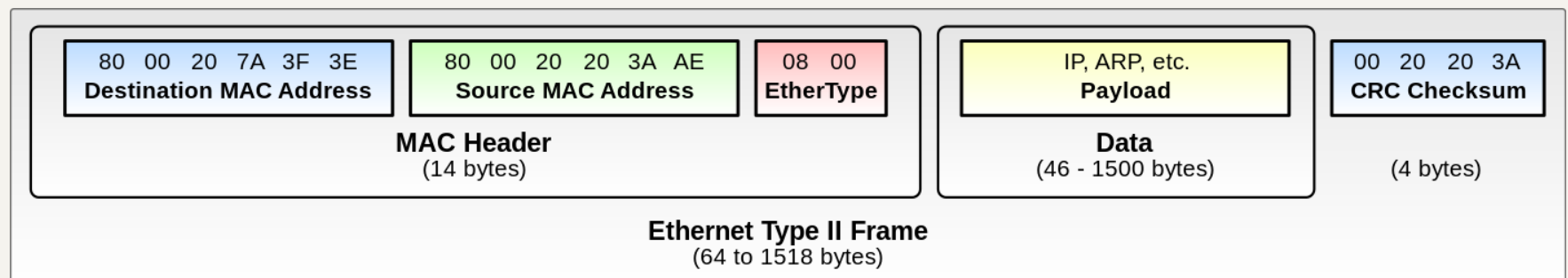


Fig 1.10 - Trame Ethernet type II

3 - Le modèle de référence OSI

Les sept couches du modèle OSI

Couche liaison de données (suite)

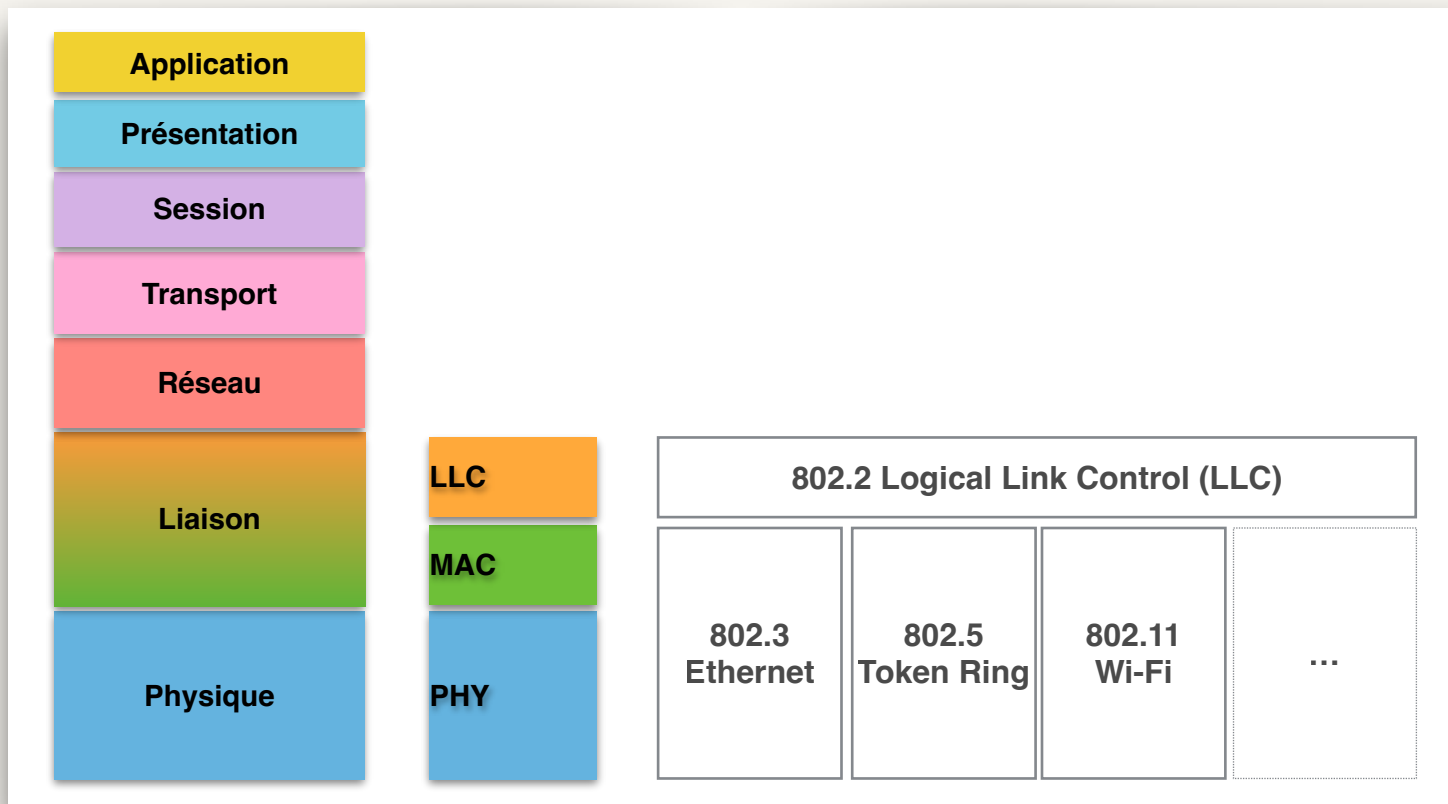
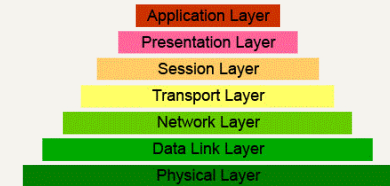


Fig 1.11 - Couches PHY, MAC et LLC



3 - Le modèle de référence OSI

Les sept couches du modèle OSI

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

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

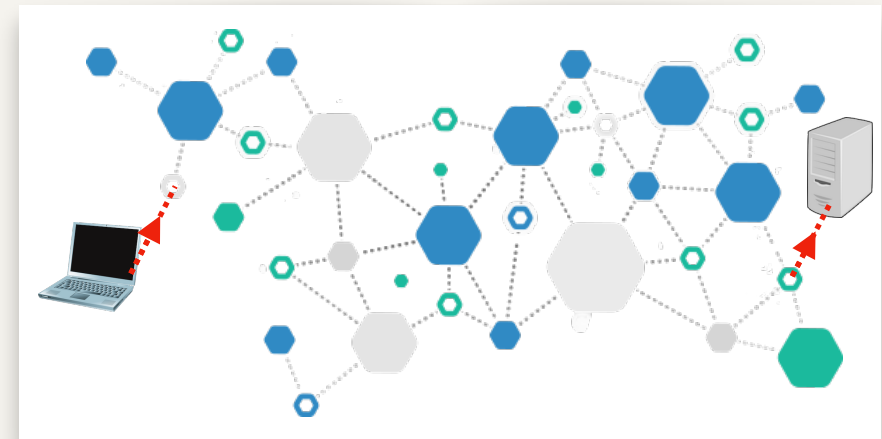


Fig 1.12 - Système de relais

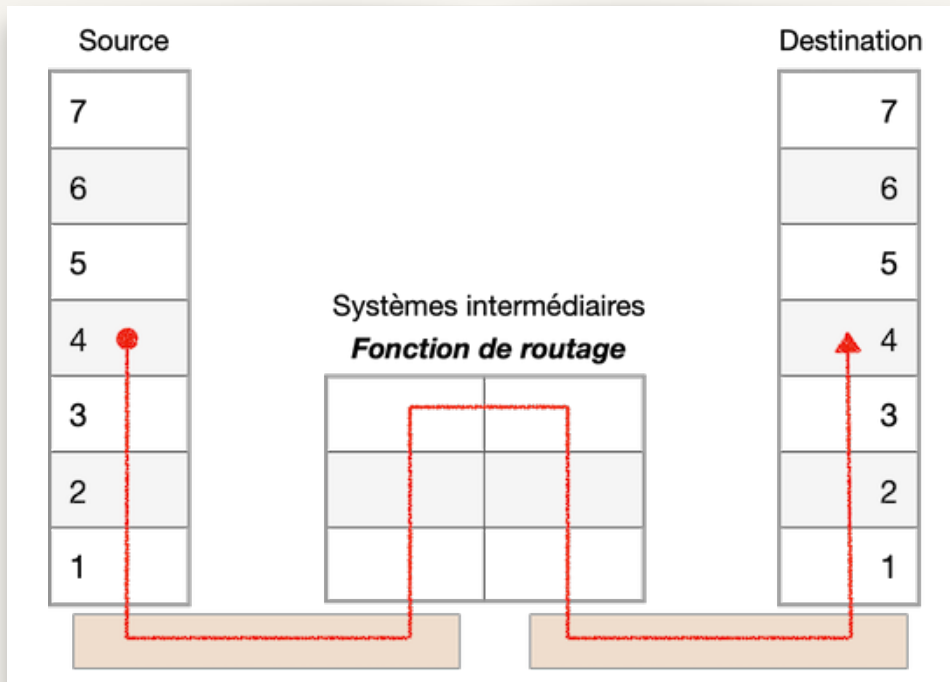
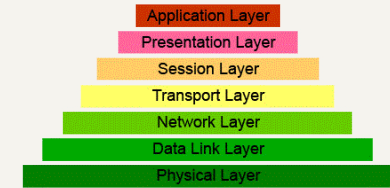


Fig 1.13 - Fonction de roulage



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.

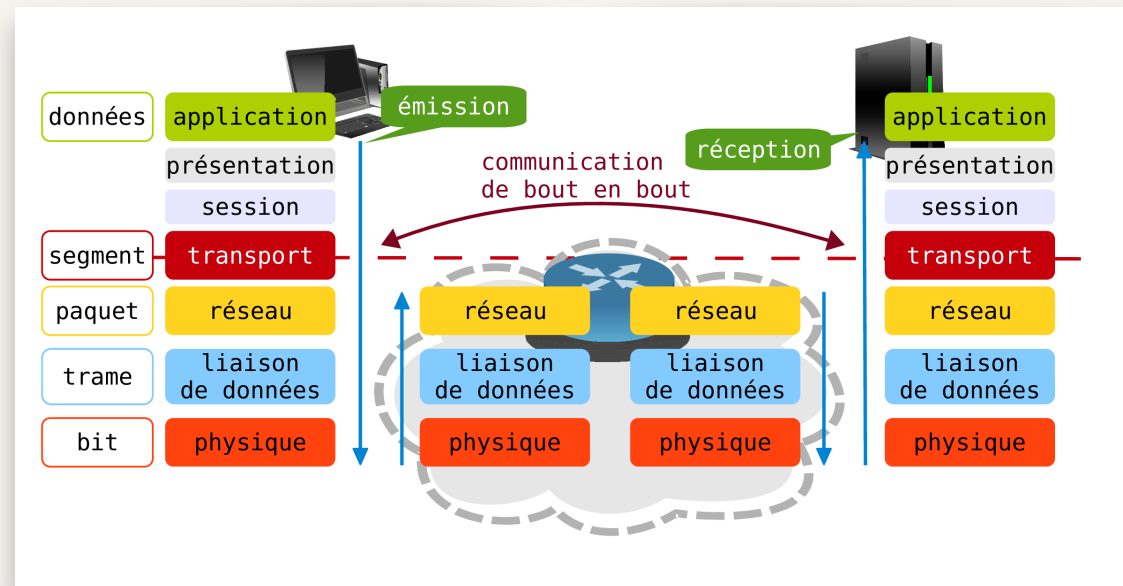
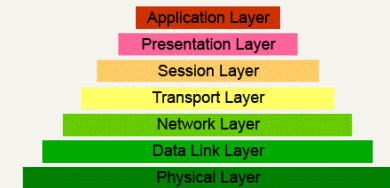


Fig 1.14 - Place de la couche transport dans le modèle OSI



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.

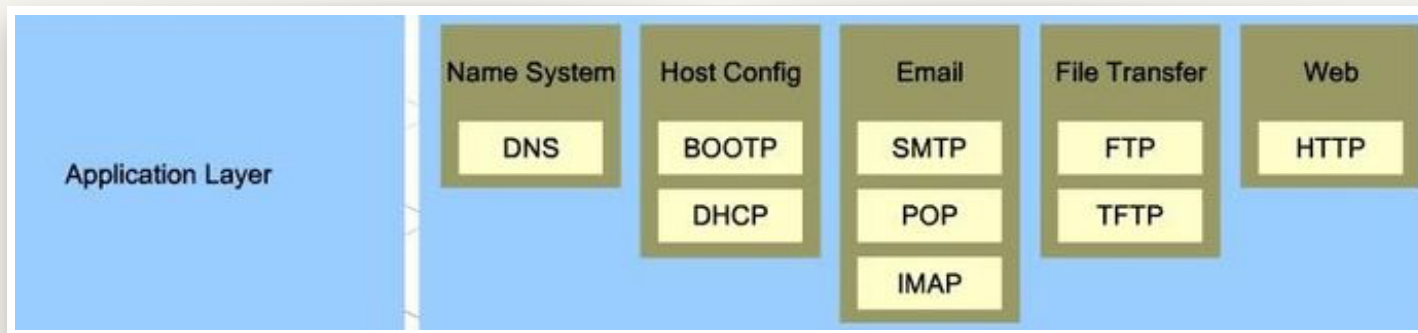
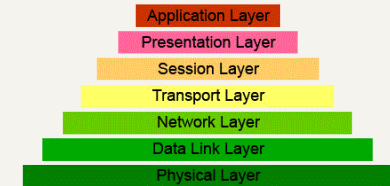


Fig 1.15 - Quelques protocoles de la couche application



Internet

ARPANET - Internet

Une architecture [ou un modèle ?] à 4 couches

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

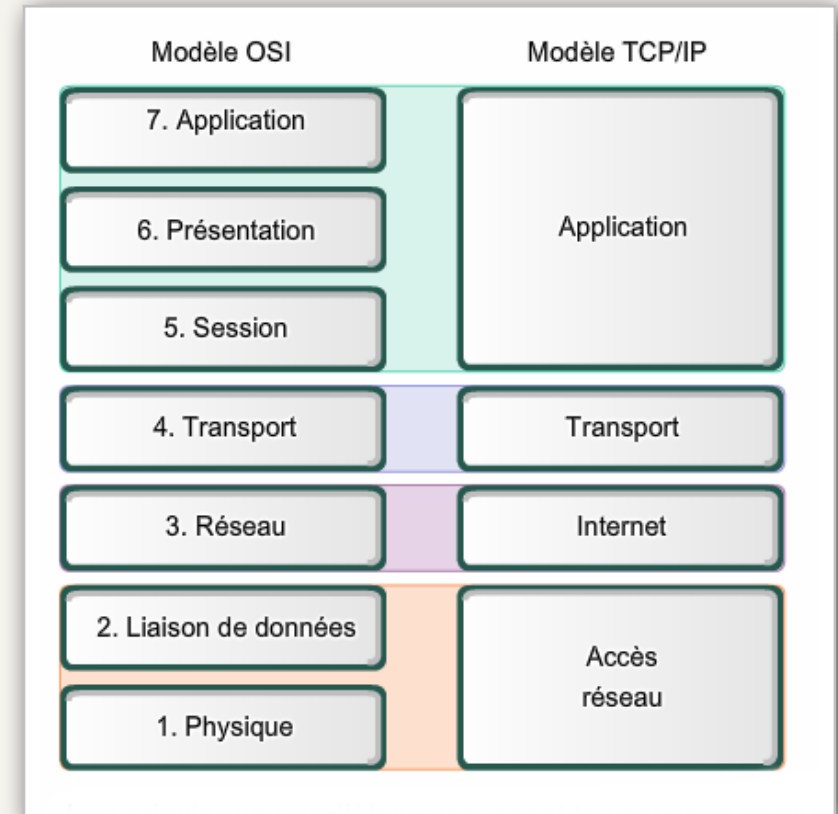
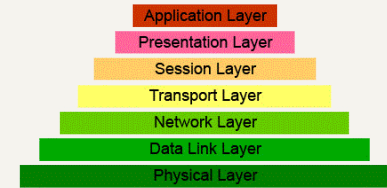


Fig 1.16 - Comparaison du modèle OSI et de l'architecture TCP/IP



4 - L'architecture TCP/IP

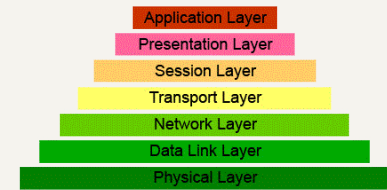
Les couches de l'architecture TCP/IP

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

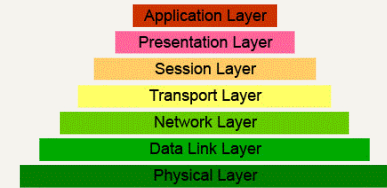


4 - L'architecture TCP/IP

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.

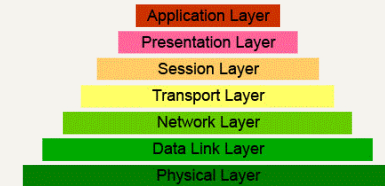


4 - L'architecture TCP/IP

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.



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 ?

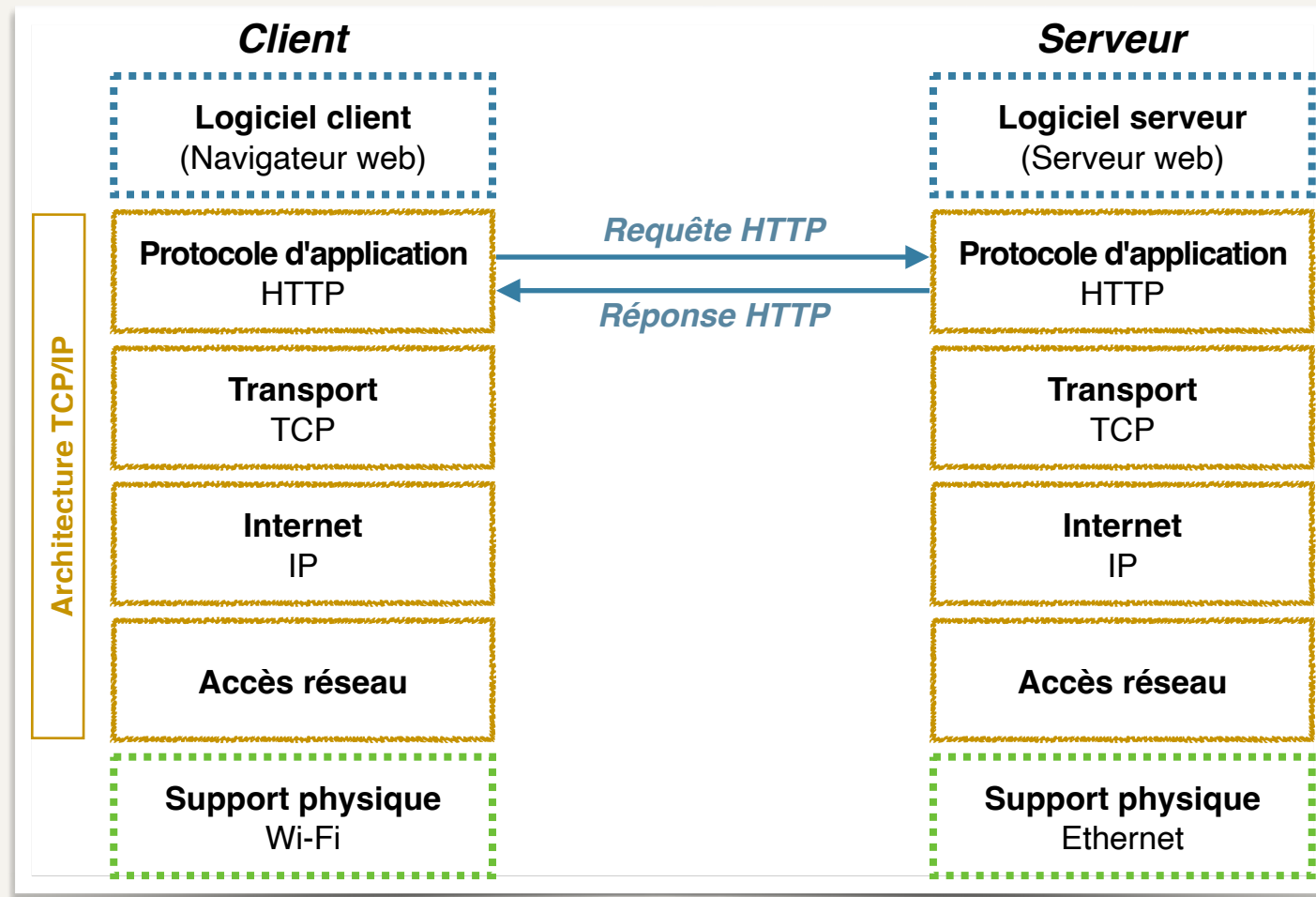
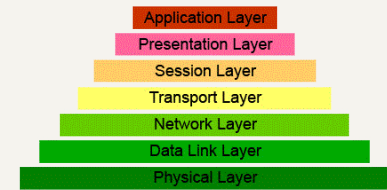


Fig 1.17 - Requête d'une page web

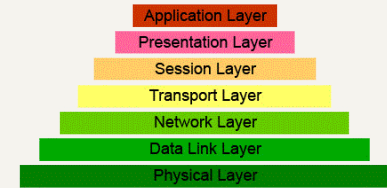


4 - L'architecture TCP/IP

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*) ;

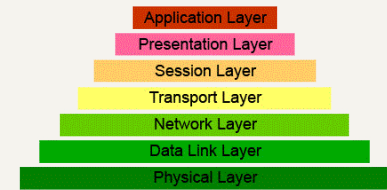


4 - L'architecture TCP/IP

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 crée les trames Wi-Fi nécessaires ;
- ▶ Qui seront codées 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

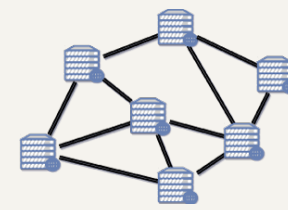


4 - L'architecture TCP/IP

La couche application

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

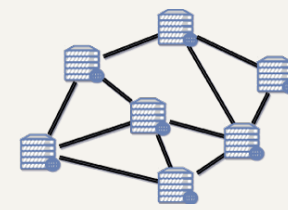
- ▶ Sur le serveur, **frames**, puis **datagramme** IP, puis **segment** TCP sont décapsulés pour fournir au processus HTTP la requête *GET/HTTP/1.1...*
- ▶ HTTP
 - analyse la requête,
 - recherche la ressource
 - l'inclue dans la réponse HTTP : *HTTP/1.1 200 OK...*
- ▶ Cette réponse sera transmise au 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

❖ *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

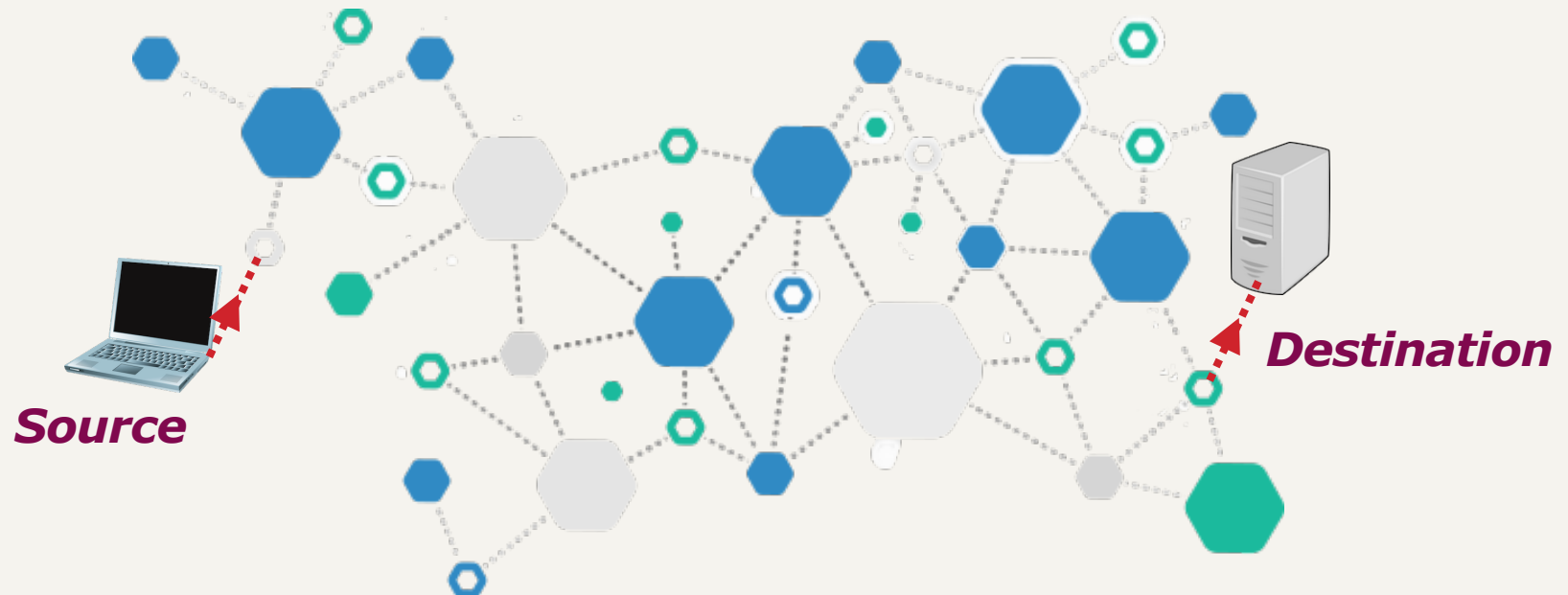
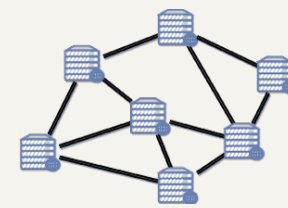


Fig 2.1 - Réseau maillé

La couche réseau

1 - Objectif de la couche réseau



Ch.2

Services de la couche réseau

Ils sont fournis à la couche transport

Couche transport

Couche réseau

Client (utilisateur)

Opérateur réseau (transporteur)

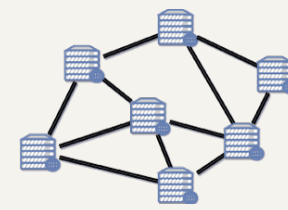
Fig 2.2 - Service de la couche réseau

Types de service :

	Mode	
	non connecté	connecté
non fiable	<i>Mode datagramme</i> 1	2
fiable	3	<i>Circuit virtuel</i> 4

Fig 2.3 - Types de service

La couche réseau

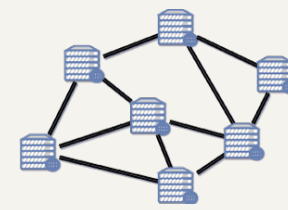


Ch.2

1 - Objectif de la couche réseau

Services de la couche réseau

Critère	Mode non fiable sans connexion	Mode fiable orienté connexion
Fonctionnement	Paquets indépendants	Route établie à l'avance
Routage	Chaque paquet peut suivre une route différente	Tous les paquets suivent le même chemin
Avantage	Flexible et tolérant aux pannes	Rapide et efficace
Inconvénient	Traitement plus complexe	Sensible aux pannes de la route
Analogie	Service postal	Service téléphonique
Exemple	Protocole IP de la couche internet	ATM, relais de trame, MPLS



Définitions ; introduction

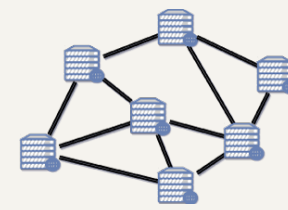
Chemin : suite de liens et de nœuds intermédiaires parcourus pour aller de la source à la destination dans le réseau

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

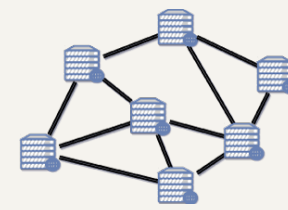
La **table de routage** est utilisée pour la fonction de **relayage** de paquets

- 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

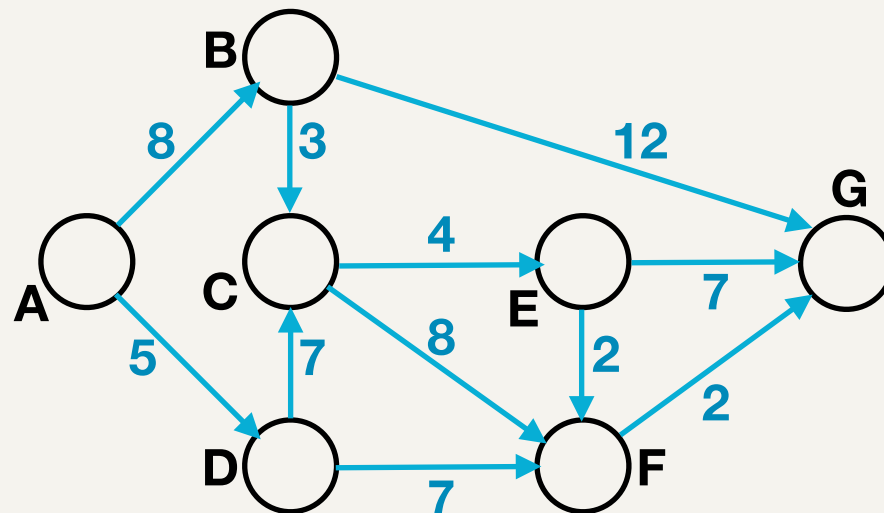
Voir : <https://www.youtube.com/watch?v=EPo7QyB7Yss>

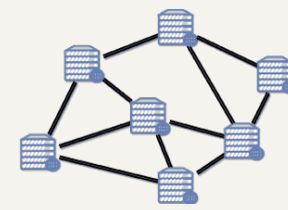


Exemples de routage

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 ?

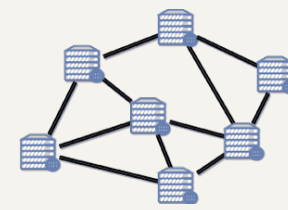




Exemples de routage

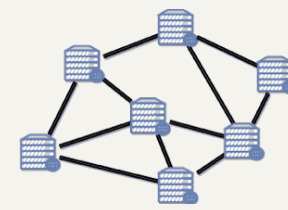
Routage à vecteur de distance - *Distance Vector Routing*

- 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

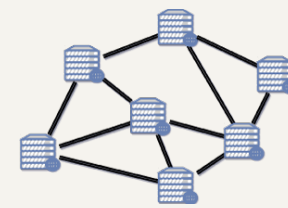
- Voir www.youtube.com/watch?v=kzablGaqUXM
- Ce routage à vecteur de distance doit être amélioré pour assurer une convergence plus rapide et pour éviter la création de boucle dans le réseau.
- On utilise pour cela la technique de l'horizon coupé (*Split Horizon*)



Exemples de routage

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

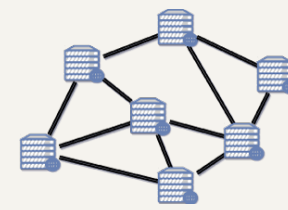
- Chaque routeur R doit :
 - Découvrir ses voisins ; un voisin V $\Rightarrow$ un lien R- $\rightarrow$ V
 - Déterminer la distance de chaque voisin : d_{RV}
 - Construire un paquet d'**information d'état de lien** [R , V , d_{RV}]
 - À chaque changement significatif, R ne diffuse que les modifications d'information d'état de liens qu'il a détecté. La diffusion concerne un sous-réseau nommé aire ou zone (*area*)
 - Chaque nœud R entretient 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
- Voir : www.youtube.com/watch?v=-utHPKREZV8



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)

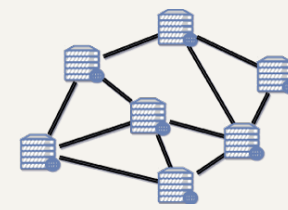
- Réseau destination : **127.0.0.0/8**
- Cette entrée gère le trafic pour le réseau de bouclage (localhost)
- Les paquets destinés à ce réseau sont acheminés vers l'interface 127.0.0.1

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



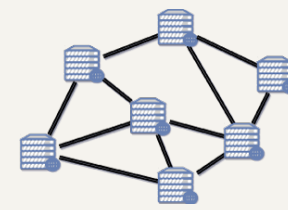
Exemple de table de routage

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

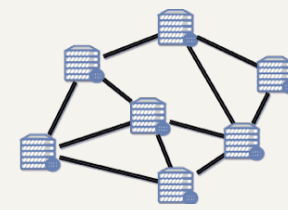
- Protocole simple parfois utilisé en Intranet
- Anciennement utilisé dans internet, mais remplacé par les protocoles ci-dessous.

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*)

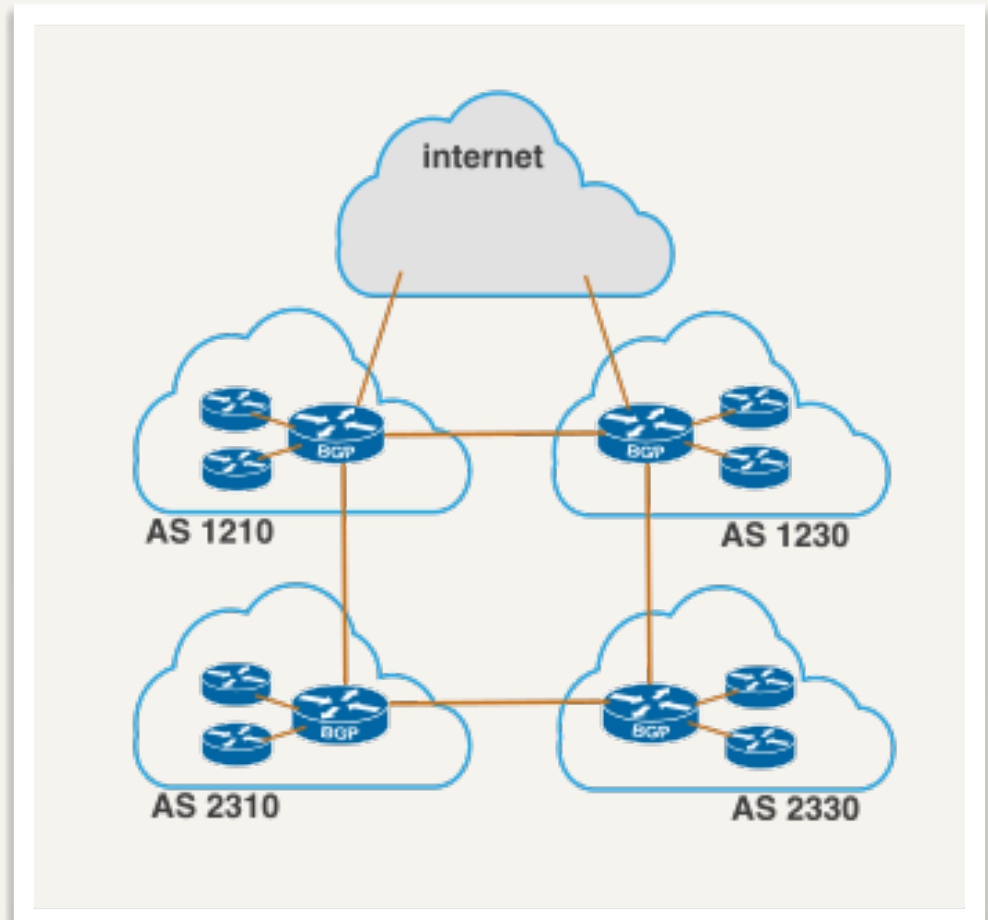
- Protocole de routage interne multi-protocoles à état de liens
- Norme ISO/CEI 10589:2002 également publié par l'IETF avec la RFC 1142
- IS-IS est un protocole à état de liens utilisé à l'intérieur d'un *autonomous system*. Il est apprécié dans des grands réseaux de fournisseurs de services.

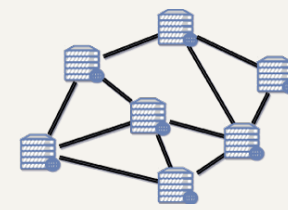


Protocoles de routage

BGP (*Border Gateway Protocol*)

- Protocole d'échange de route externe
- Il échange des 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.





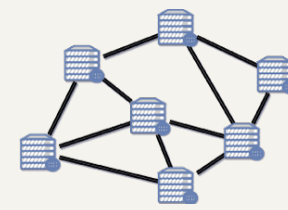
Besoins d'interconnexion

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'interconnexion

- 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

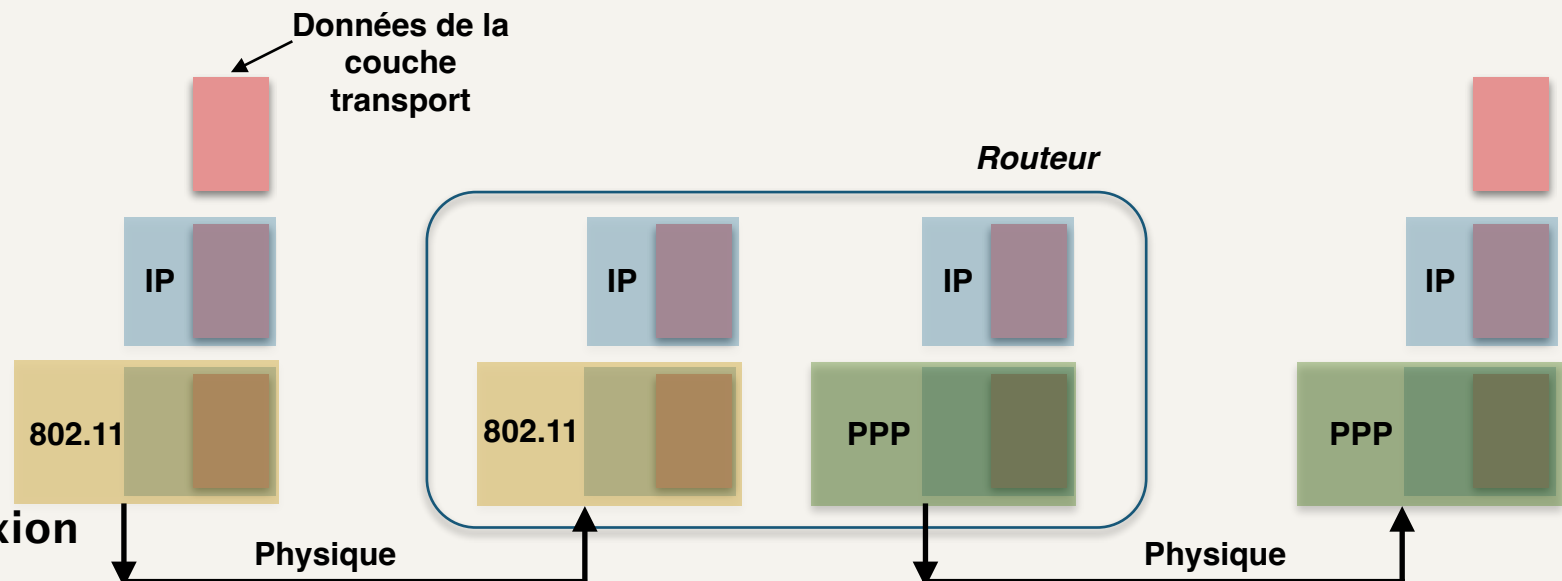
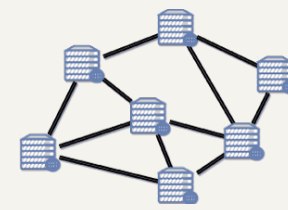


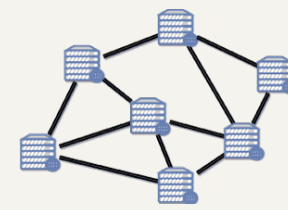
Fig 2.4 - Interconnexion avec un routeur



Équipements d'interconnexion

La technique du tunnel

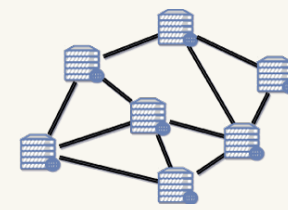
- Les machines d'extrémité sont sur un réseau de même type, mais elles sont séparées par un réseau différent
- Une solution d'interconnexion passe par la technique du tunnel ou le paquet source est encapsulé par le routeur qui ajoute son entête de niveau 3
- Cette technique est souvent utilisée au niveau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



4 - La couche Internet - Présentation

Architecture TCP/IP

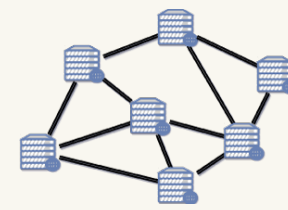
- ❖ Dans cette partie, ou presque :
 - Historique (*recherche personnelle à faire*)
 - La couche Internet ; le protocole IP
 - Les adresses IP v4
 - Sous réseaux
 - IP v6
 - Autres protocoles de la couche internet
 - La couche transport de TCP/IP (Ch. 3)



4 - La couche Internet - Présentation

Couche internet

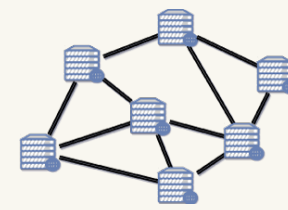
- Nommée couche *internet* ou couche *inter-réseau*
- Au même niveau de la couche **réseau** d'OSI
- Objectif :
 - Permettre aux hôtes d'introduire des paquets nommés datagrammes sur n'importe quel réseau
 - Acheminer ces datagrammes indépendamment les uns des autres jusqu'à destination.
- Des datagrammes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



4 - La couche Internet - Le protocole IP

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



4 - La couche Internet - Le protocole IP

L'en-tête du datagramme IPv4

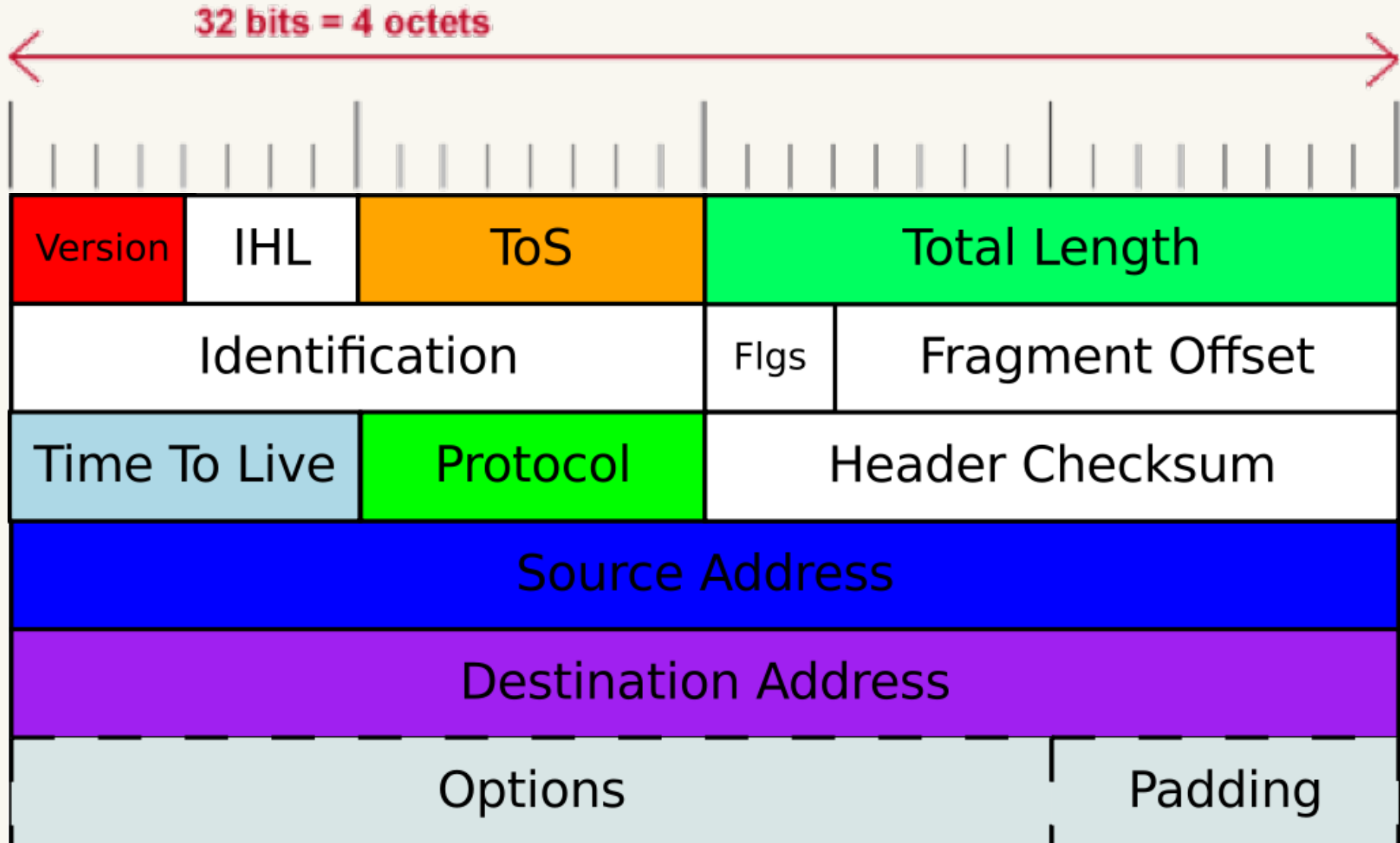
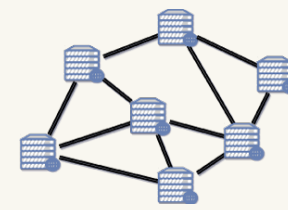


Fig 2.5 - L'en-tête du datagramme IPv4

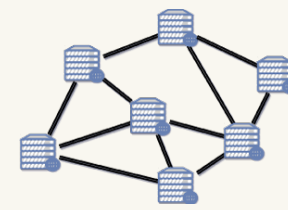


4 - La couche Internet - Le protocole IP

Le datagramme IPv4

- L'en-tête est de 20 à 60 octets
 - **Version IP (4 bits) = 4**
 - **IHL, Internet Header Length** ; longueur 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

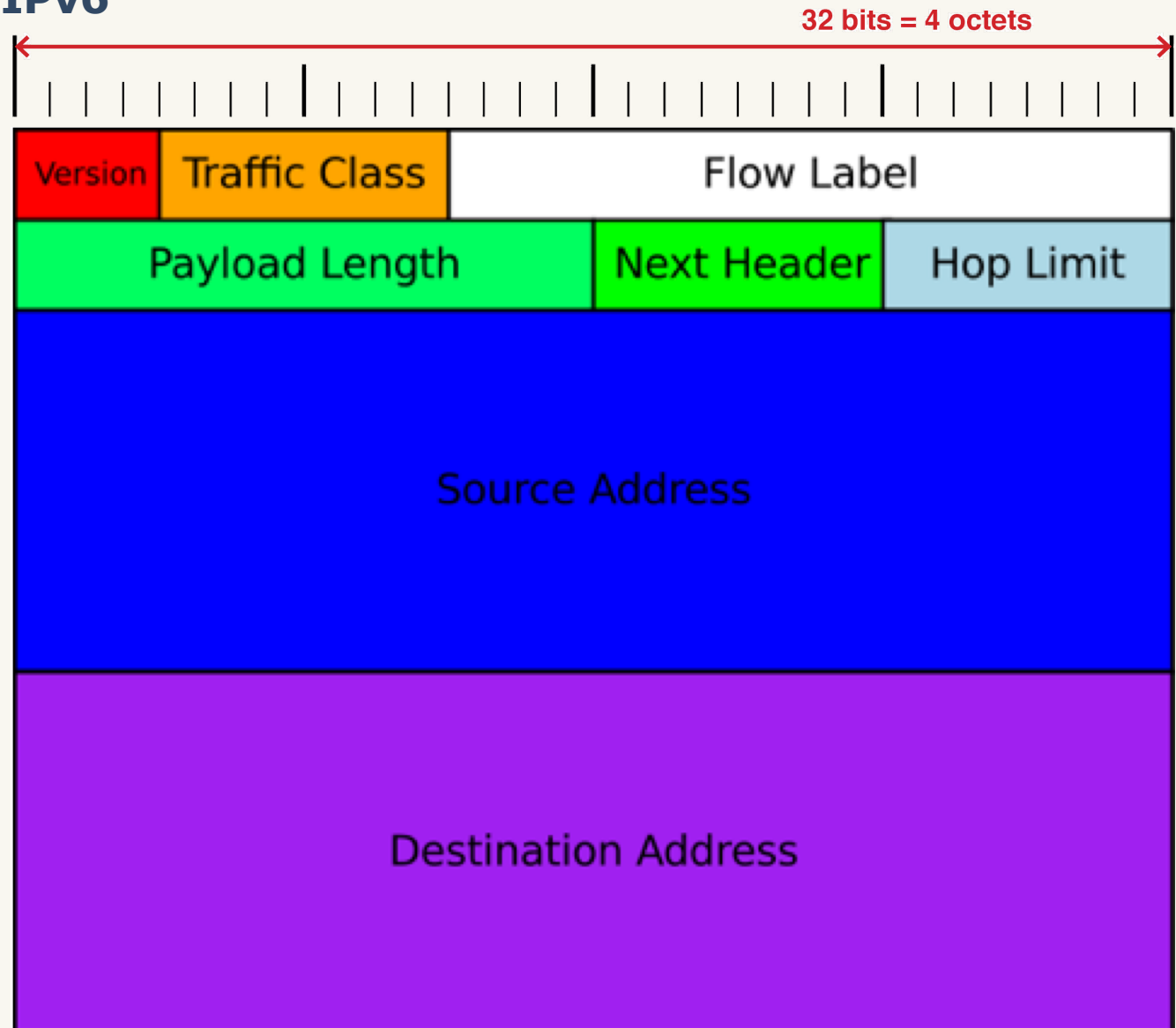
La couche réseau



Ch.2

4 - La couche Internet - Le protocole IP

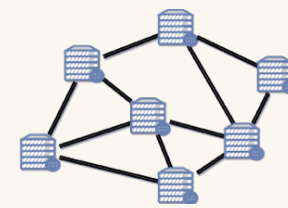
L'en-tête du datagramme IPv6



► Un en-tête IPv6 de 40 octets

Adresses IPv6 de 16 octets,
soit 128 bits

Fig 2.6 - L'en-tête du datagramme IPv6



4 - La couche Internet - Le protocole IP

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)

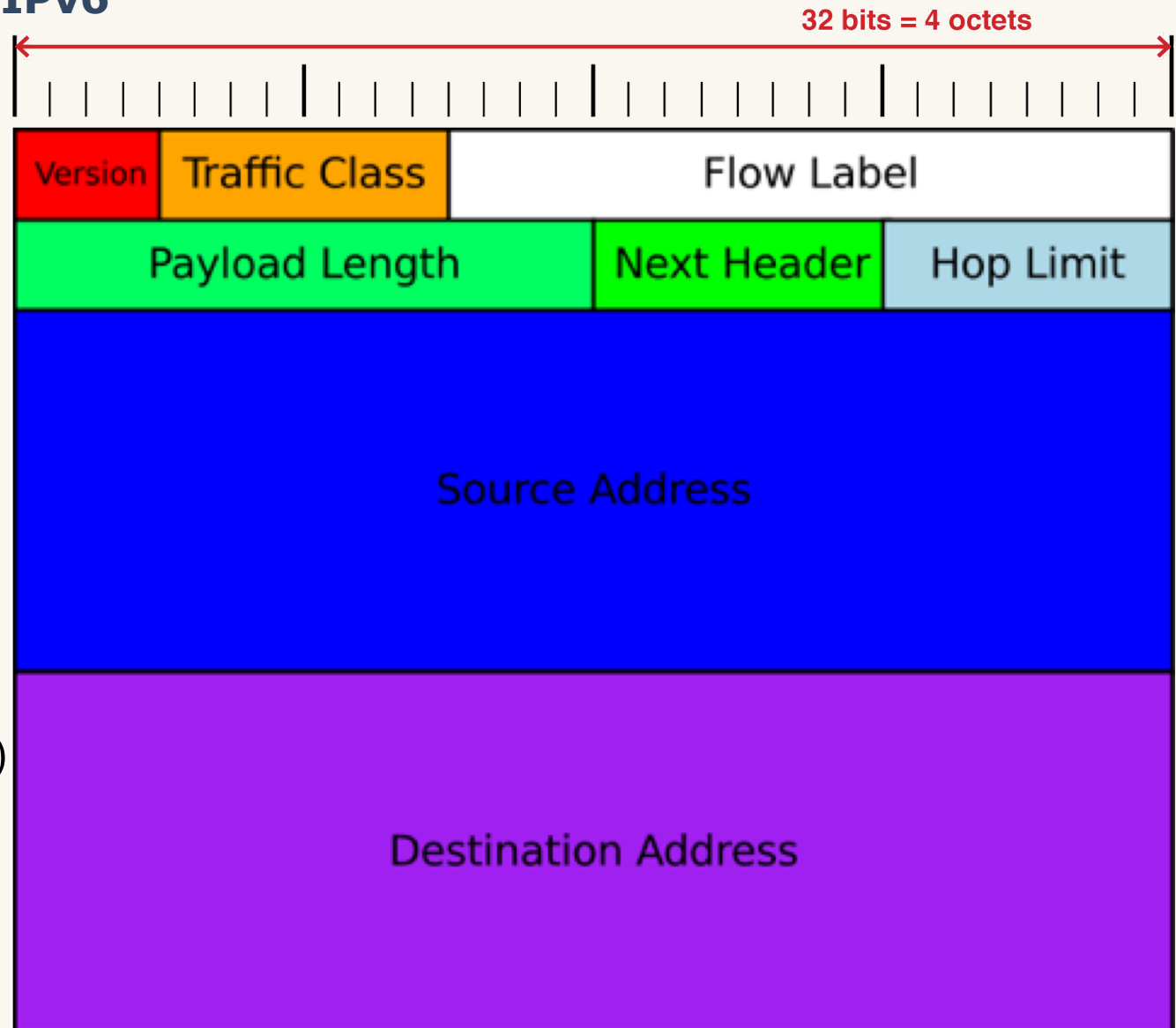
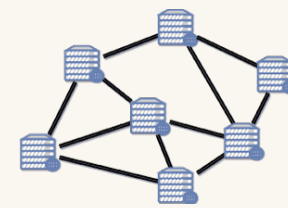


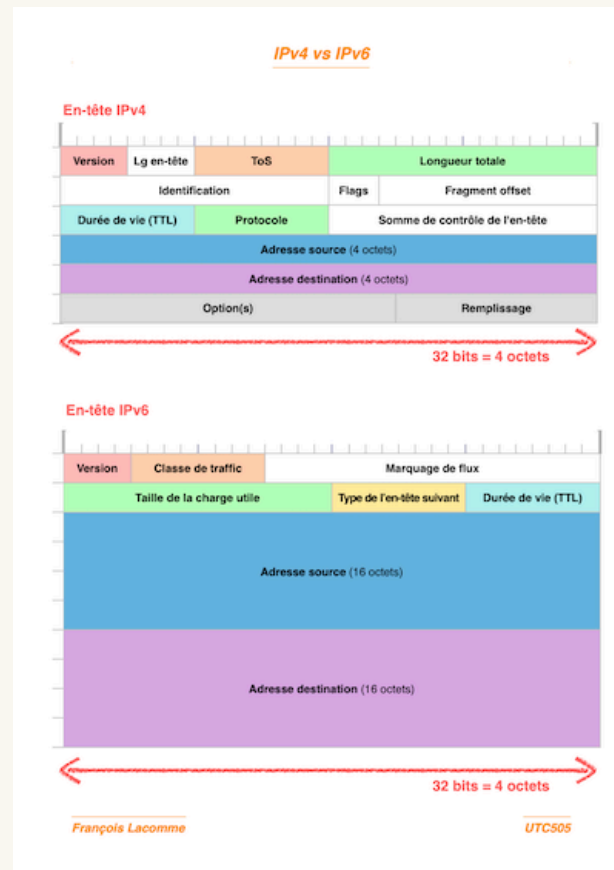
Fig 2.6 - L'en-tête du datagramme IPv6

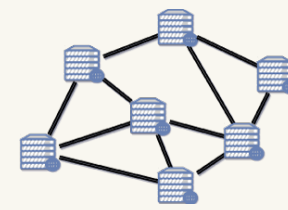


4 - La couche Internet - Le protocole IP

Comparaison des en-têtes IPv4 et IPv6

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

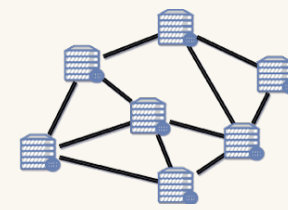




4 - La couche Internet - Les adresses IPv4

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



4 - La couche Internet - Les adresses IPv4

Exemple 1

172 . 16 . 254 . 1

↓ ↓ ↓ ↓

10101100.00010000.11111110.00000001

└───┬───┘ └───┬───┘

1 octet = 8 bits

└──┘

32 bits (4 * 8), ou 4 octets

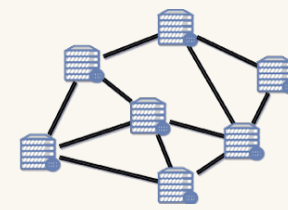
Exemple 2

191 . 32 . 127 . 15

↓ ↓ ↓ ↓

10111111.00100000.01111111.00001111

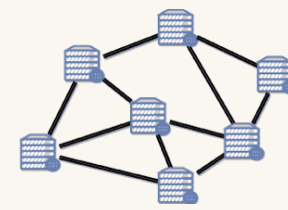
Fig 2.7 -Deux exemples de conversion d'adresse IP (décimal vers binaire)



4 - La couche Internet - Les adresses IPv4

Structure d'une adresse IPv4

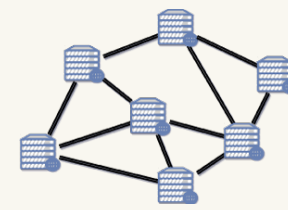
- Une adresse IPv4 permet d'identifier :
 - un **réseau**
 - l'**interface réseau** d'un équipement sur **ce réseau** [on parle, improprement, d'hôte (*host*)]
- Pour déterminer l'**identifiant réseau**, on doit connaître le **masque d'adresse**
- Un masque d'adresse est **une suite de bits à 1** suivie d'**une suite de bits à 0** (le tout sur 32 bits)



4 - La couche Internet - Les adresses IPv4

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)
- Exemple :
 - La notation CIDR **191. 32.127.15/20** correspond à une adresse **191. 32.127.15** associé au masque **255.255.240.0**
- Un **ET logique** ($\wedge$) entre une adresse et un masque permet de déterminer l'**identifiant réseau**
- L'adresse IPv4 $\wedge$ le complément à un du masque détermine l'**identifiant d'une interface-réseau** (ou, improprement, l'identifiant d'hôte [*host ID*])



4 - La couche Internet - Les adresses IPv4

❖ Exemple 1

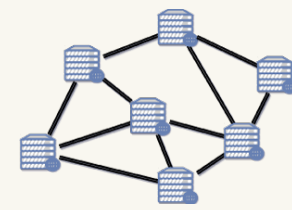
Identifiant réseau

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



4 - La couche Internet - Les adresses IPv4

❖ Exemple 2

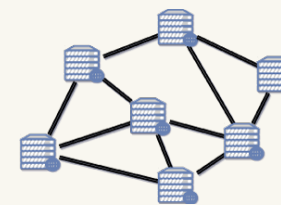
▸ Identifiant réseau

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



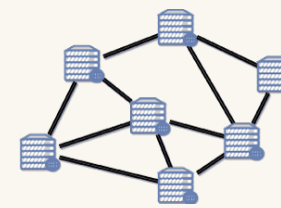
4 - La couche Internet - Les adresses IPv4

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 2.10 - Tableau de 5 classes d'adresse



4 - La couche Internet - Les adresses IPv4

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*)

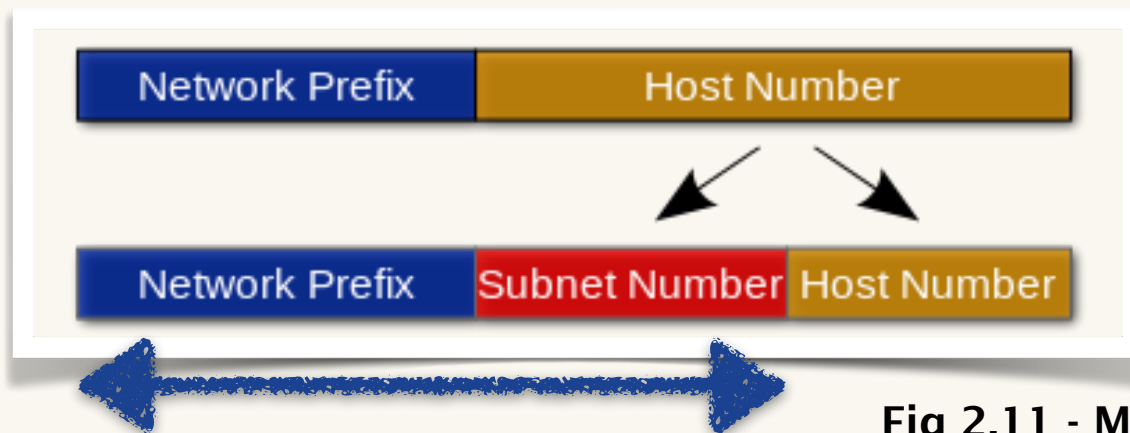
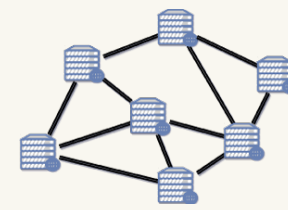


Fig 2.11 - Masque de sous-réseau

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

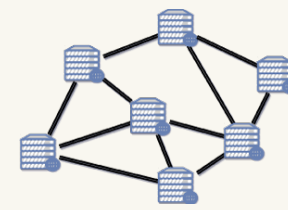


4 - La couche Internet - Les adresses IPv4

Exemples

❖ Exemple 1

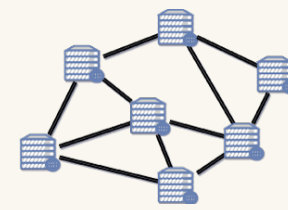
- Une entreprise dispose du bloc d'adresse : 197.201.30.0 / 24
 - Le masque associé est donc : 255.255.255.0
- On souhaite créer 16 sous-réseaux :
 - Le masque de sous-réseau voit sa longueur augmenter de 4 (car $2^4 = 16$)
 - ce masque devient donc 255.255.255.240
 - car $240 \equiv 11110000_b$
 - (on 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 4^e octet donne l'adresse de sous-réseau
 - les quatre bits de poids faible du 4^e octet donne l'identifiant d'hôte dans le sous-réseau



4 - La couche Internet - Les adresses IPv4

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 1^{er} sous-réseau) et 197.201.30.15, qui est utilisé pour une diffusion dans ce sous-réseau.
 - ▶ Comment s'écrit le 2^e sous-réseau ?
 - ➔ 197.201.30.16 / 28



4 - La couche Internet - Les adresses IPv4

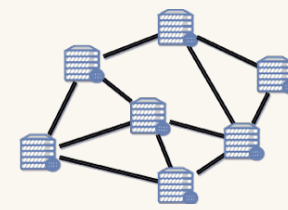
Exemples

❖ 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



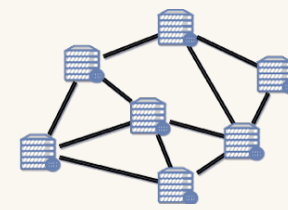
4 - La couche Internet - Sous-réseau IPv4

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
$S \wedge M$	193. 49. 66.192	1100'0000
$D \wedge M$	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



4 - La couche Internet - Sous-réseau IPv4

Exemples

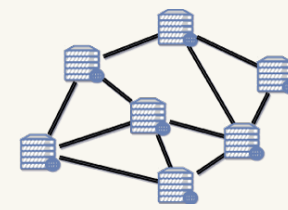
❖ (Suite exemple 2)

► On peut aussi 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
$S \wedge M$	193. 49. 66.192	193. 49. 66.1100'0000
$D \wedge M$	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



4 - La couche Internet - Sous-réseau IPv4

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*)



4 - La couche Internet - Adresses particulières ; adressage privé

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 ($\approx$ *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



4 - La couche Internet - Adresses particulières ; adressage privé

Adressage privé

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

<u>Préfixe</u>	<u>Plage d'adresse</u>	<u>Nombre d'adresses</u>
▸ 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*)



4 - La couche Internet - Adresses particulières ; adressage privé

NAPT (Network Address and Port Translation)

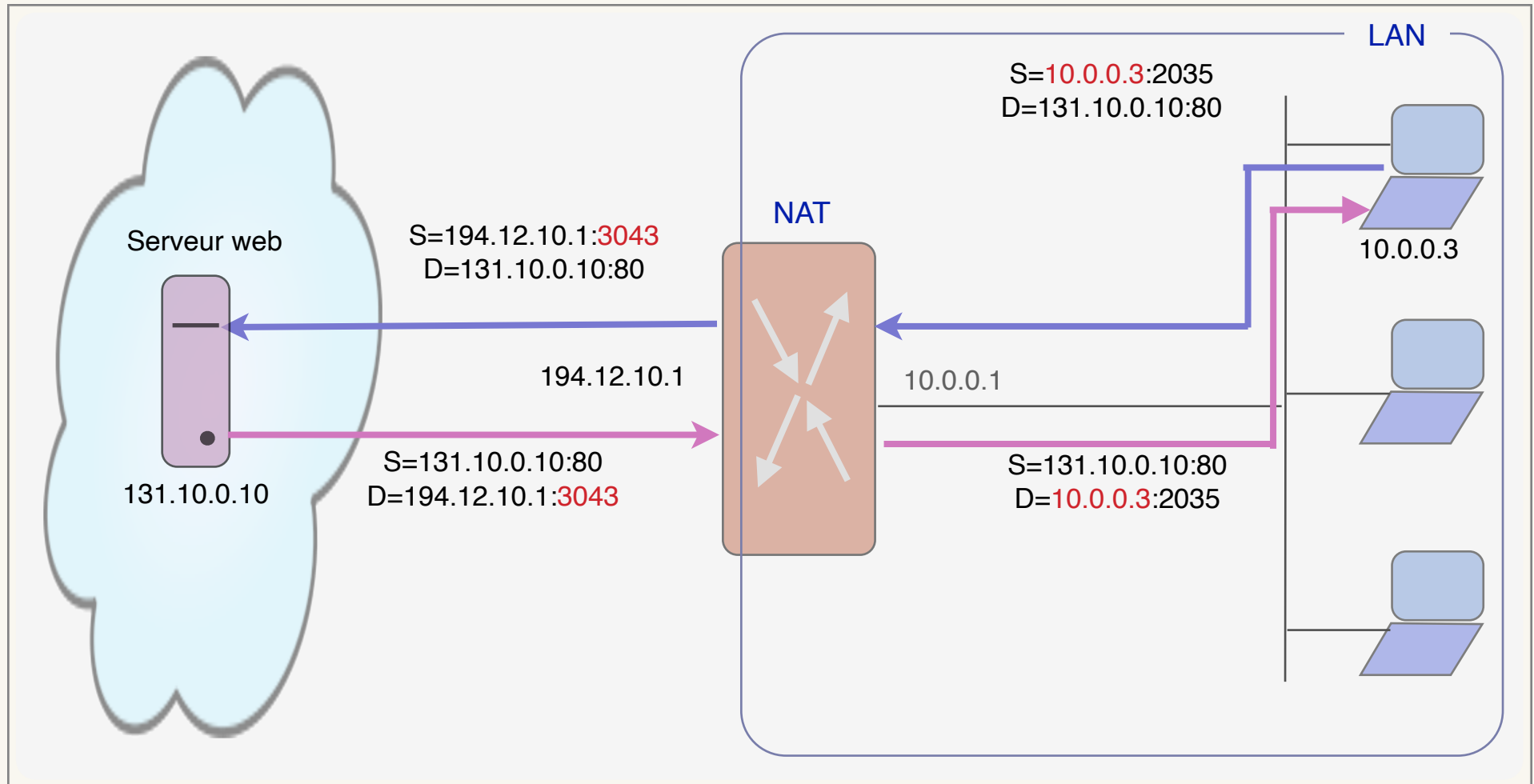
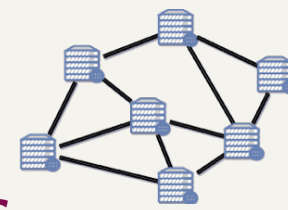


Fig 2.12 -Fonction NAPT d'un routeur



4 - La couche Internet - Configuration IPv4 des hôtes

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

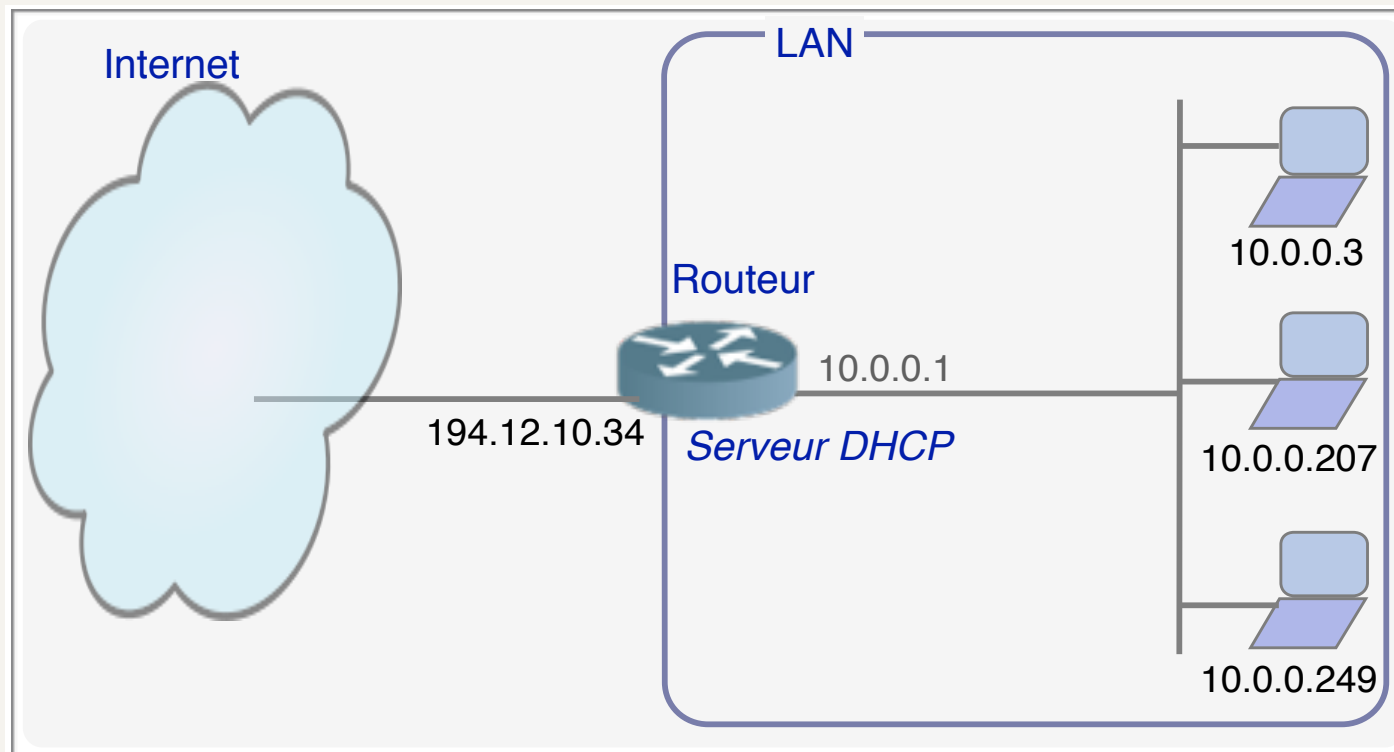
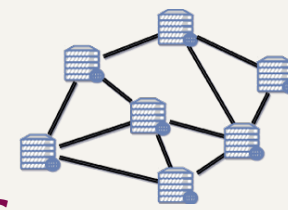


Fig 2.13 - Serveur DHCP associé à un routeur



4 - La couche Internet - Configuration IPv4 des hôtes

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) à modifier, puis éditez les propriétés de l'élément « *Protocole Internet version 4 (TCP/IPv4)* ».

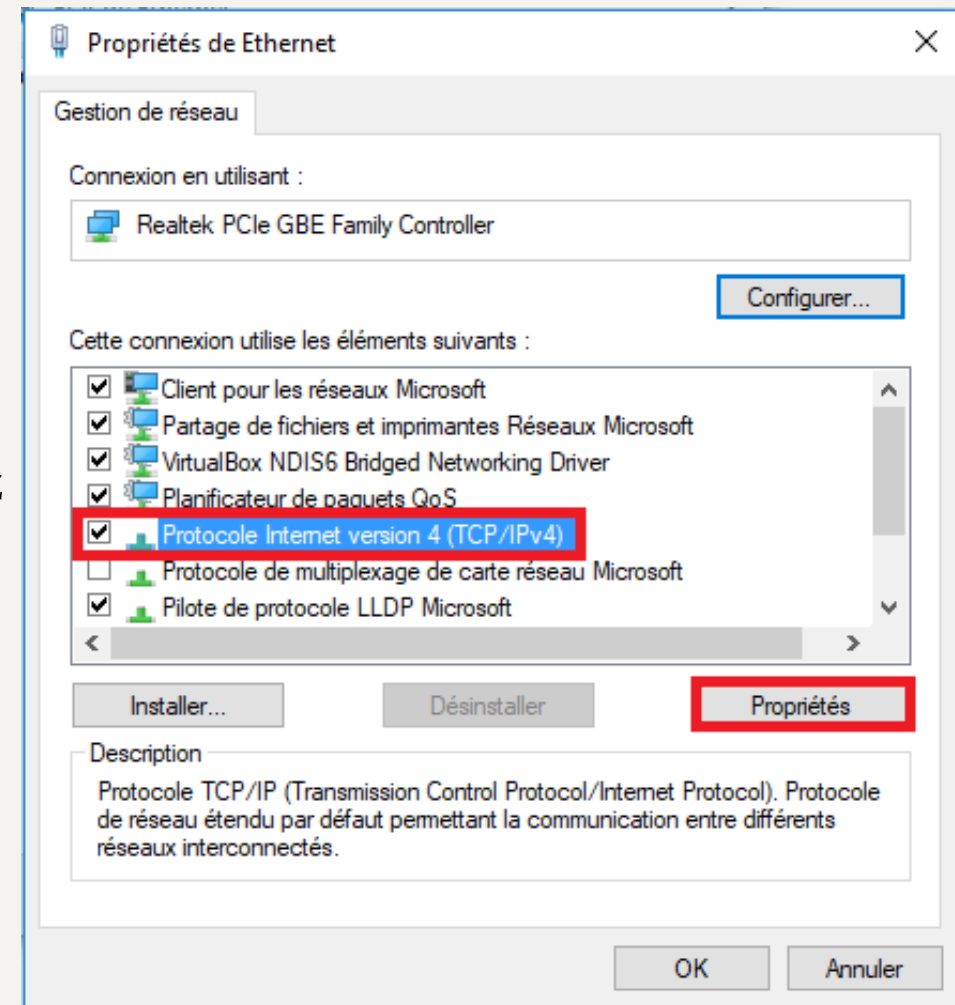
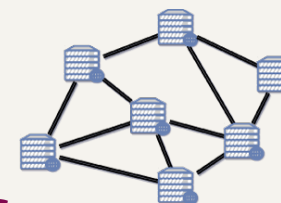


Fig 2.14 - Propriétés de 'Protocole Internet v4' sous Windows



4 - La couche Internet - Configuration IPv4 des hôtes

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) à 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.

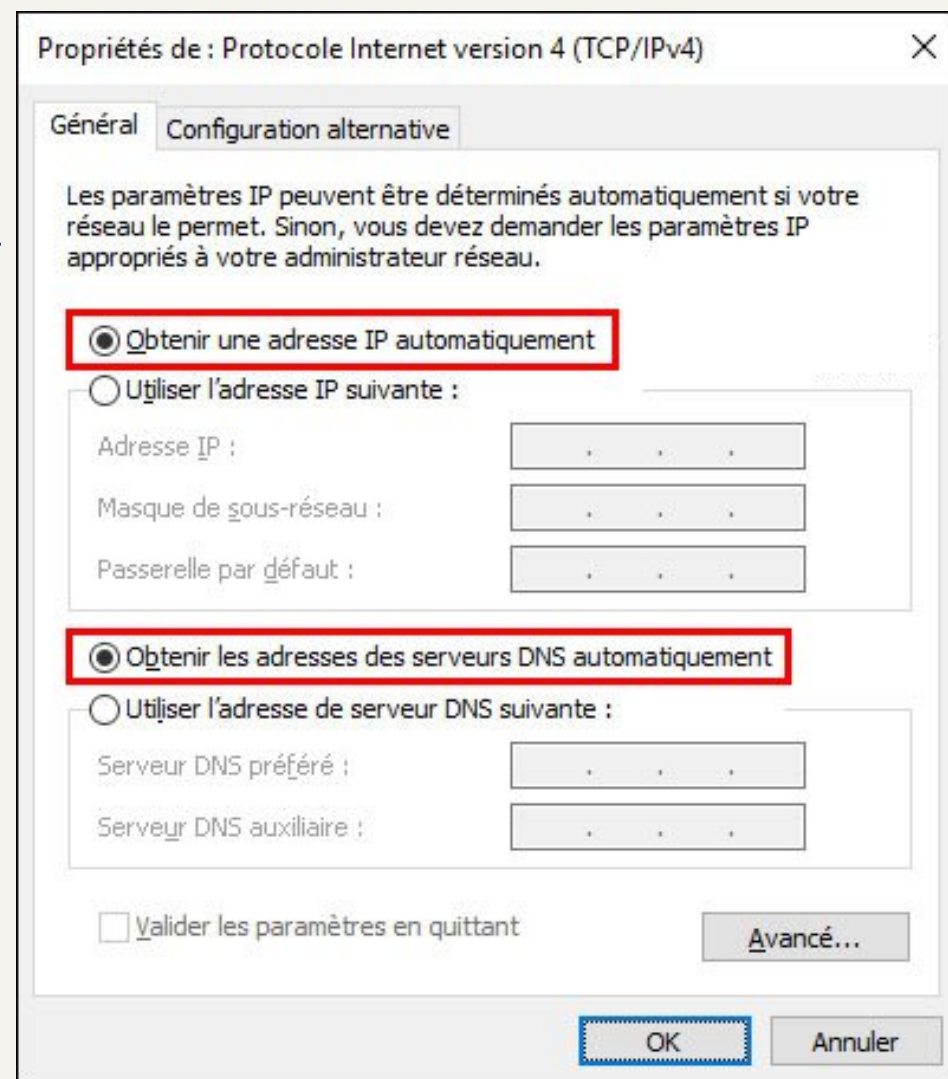
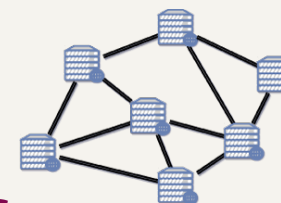


Fig 2.14 - Propriétés de 'Protocole Internet v4' sous Windows



4 - La couche Internet - Configuration IPv4 des hôtes

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) à 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.
- ❖ Pour Windows 11, voir aussi :
 - Lien : www.malekal.com/configurer-une-adresse-ip-sous-windows-11-10-dhcp-ou-ip-fixe/

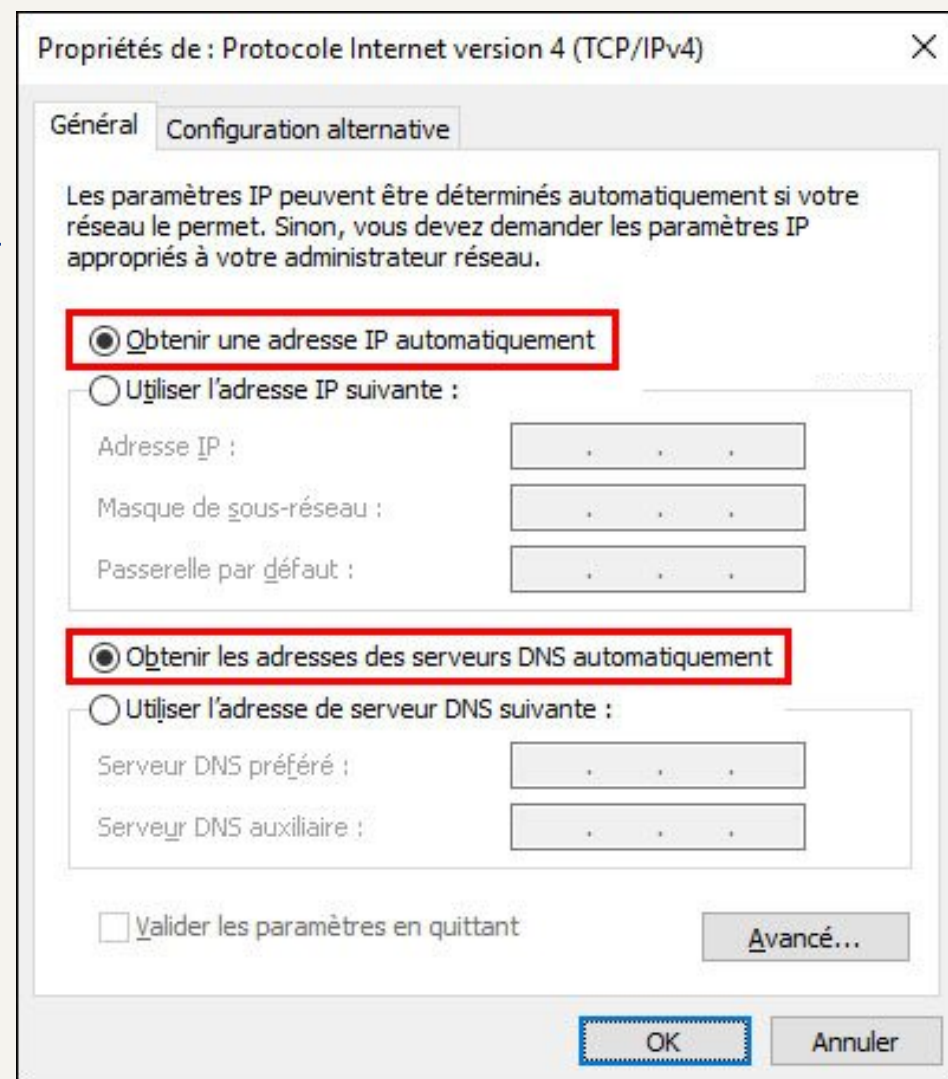
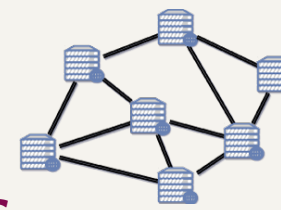


Fig 2.14 - Propriétés de 'Protocole Internet v4' sous Windows



4 - La couche Internet - Configuration IPv4 des hôtes

Configuration dynamique

- ❖ Avec macOS
 - ▶ Lien : support.apple.com/fr-fr/guide/mac-help/mchlp2718/mac
 - ▶ Réglages système / Réseau  / Sélectionner un service de connexion puis cliquer sur *Détails...*
 - ▶ Choisir l'onglet TCP/IP, puis avec le menu déroulant « Configurer IPv4 », choisir « Via DHCP »

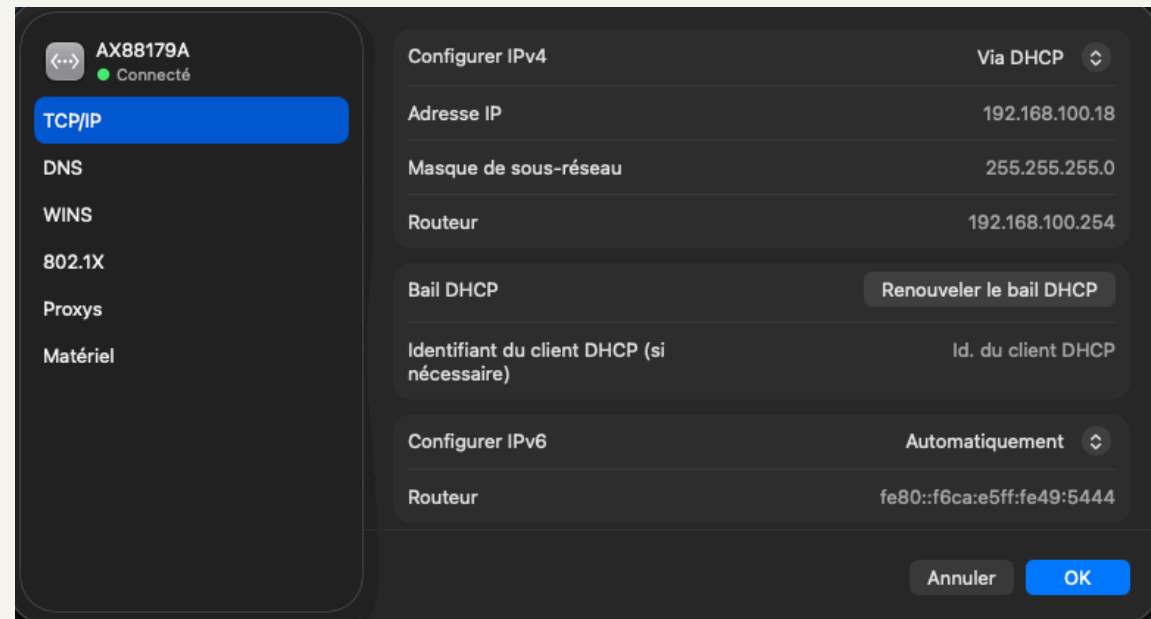
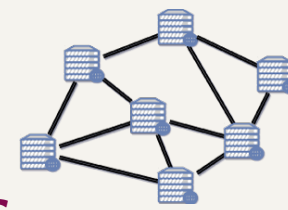


Fig 2.15 - Réglages d'un service de connexion réseau sous MacOS

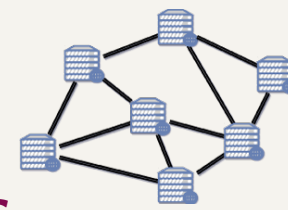


4 - La couche Internet - Configuration IPv4 des hôtes

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
 - ...

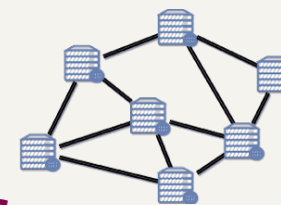


4 - La couche Internet - Configuration IPv4 des hôtes

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



4 - La couche Internet - Configuration IPv4 des hôtes

Commandes utiles

- ❖ Windows

- **ipconfig**

```
C:>ipconfig
```

```
Configuration IP de Windows
```

```
Carte réseau sans fil Connexion réseau sans fil :
```

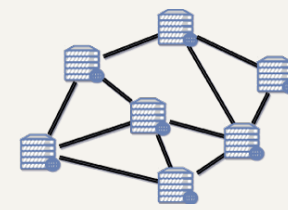
```
Statut du média. . . . . : Média déconnecté  
Suffixe DNS propre à la connexion. . . :
```

```
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**

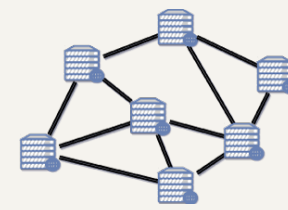
- Permet d'avoir toutes les caractéristiques des connexions réseaux : adresse IP, adresse MAC...



4 - La couche Internet - Autres protocoles

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 utilisés. 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)



4 - La couche Internet - Autres protocoles

ICMP - Internet Control Message Protocol

- * La commande **ping** utilise ICMP pour mesurer des RTT, *round-trip time*.
 - RTT est le temps, mesuré en millisecondes, qu'un paquet met pour aller d'un point de départ à une destination et revenir
- * **ping** mesure, entre autres, le délai entre l'envoi d'une requête ICMP '*echo-request*' et la réponse '*réponse echo*'.
 - Cela permet d'évaluer la latence réseau.

▸ `$ 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

RTT

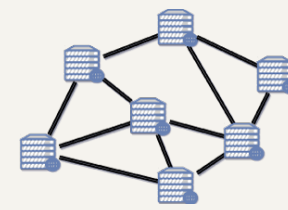


RTT moyenne



écart type RTT

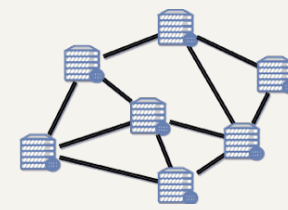




4 - La couche Internet - Autres protocoles

ARP - Address Resolution Protocol

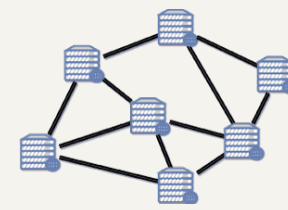
- ❖ ARP permet de trouver l'adresse physique (adresse MAC) d'un hôte à partir d'une adresse IP
 - RFC 826
 - En chaque station, ARP maintient un cache ARP
 - A veut émettre une trame Ethernet vers un ordinateur B d'adresse IP *ad-ip-b* :
 - Si, pour *ad-ip-b*, aucune adresse physique n'est indiqué dans 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.



4 - La couche Internet - Autres protocoles

Les protocoles de routage

- ❖ Les protocoles suivants font partie de la couche Internet de l'architecture TCP/IP. Nous les avons déjà décrits
 - ▶ RIP (*Routing Information Protocol*)
 - ▶ IS-IS (*Intermediate system to intermediate system*)
 - ▶ OSPF (*Open Shortest Path First*)
 - ▶ BGP (*Border Gateway Protocol*)

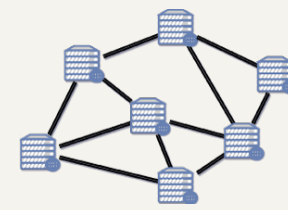


4 - La couche Internet - Autres protocoles

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 :
 - **$D \wedge M == ND \wedge M$** avec **ND** *Network destination*, **M** *Netmask*
 - Exemple pour 192.168.0.27 :
 $192.168.0.27 \wedge 255.255.255.0 == 192.168.0.0 \wedge 255.255.255.0$

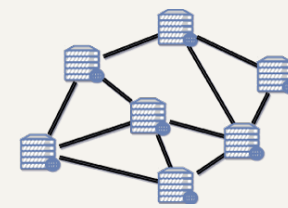


4 - La couche Internet - Autres protocoles

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

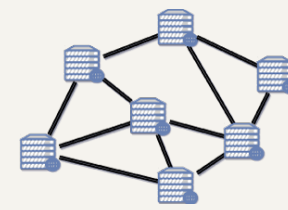


4 - La couche Internet - IPV6

État des lieux

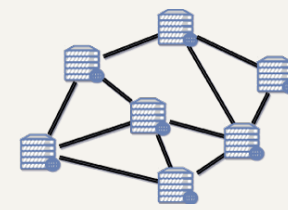
- ❖ En prévision d'un manque d'adresses IPv4, l'IETF (*Internet Engineering Task Force*) émet en 1990 un appel à proposition (RFC 1550)
 - 3 propositions sur 21 sont publiées, dont **SIPP** (Simple IP Plus), devenu **IPv6**
 - IPv6 est adopté en 1998 comme successeur d'IPv4
 - RFC 2460
- ❖ IPv4, limité à 4 milliard et quelques adresses a pu survivre :
 - malgré une forte demande en Asie et pour les nouvelles applications
 - **grâce** 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

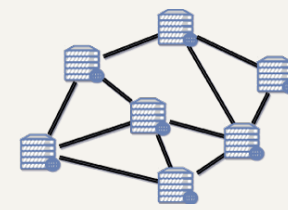
- ❖ Le nombre d'adresses (2^{128} au lieu de $\sim 2^{32}$)
- ❖ Les mécanismes de configuration et de re-numérotation automatique
- ❖ IPsec, QoS et multicast en natif
- ❖ En-têtes simplifiés, cascadables et facilitant le routage



4 - La couche Internet - IPV6

Adresses IPv6

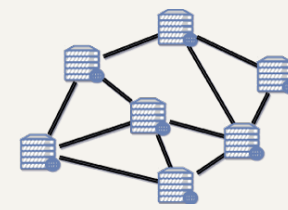
- ❖ Une adresse fait 128 bits, soit 16 octets
- ❖ On utilise une notation **hexadécimales** :
 - 8 groupes de 2 octets, séparé par « : »
 - Soit 8 groupes de 16 bits ; soit 8 groupes de 4 chiffres hexadécimaux ; soit 8 **hextets**.
- ❖ Exemple :
 - **1fff:0000:0a88:85b3:0000:0000:ac1f:8001**
- ❖ Il est 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:**



4 - La couche Internet - IPV6

Adresses IPv6

- ❖ Simplifications ou compressions (suite) :
 - ▶ 1 suite **unique** de groupes de 16 bits tous à 0 peut être omise
 - ▶ **:0000:0000:** $\Leftrightarrow$ **::**
- ❖ 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**



4 - La couche Internet - IPV6

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 $59_d = 7 \times 8 + 3 \text{ bits}$

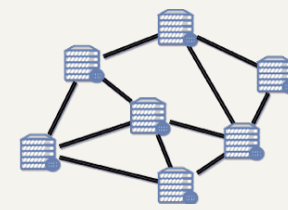
M = ffff:ffff:ffff:ffe0::

$e0_h \equiv 1110'0000_b$ (pour déterminer M)

$a0_h \equiv 1010'0000_b$

$bf_h \equiv 1011'1111_b$

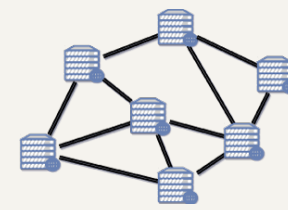
- ❖ Le préfixe **par défaut** a une longueur de 64



4 - La couche Internet - IPV6

Blocs d'adresses IPv6

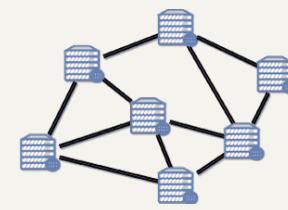
- ❖ Les adresses **2000::/3** sont les adresses routables sur internet (*Global Unicast*)
 - Quelle est la première adresse de ce bloc ? Et quelle est la dernière ?
- ❖ Les autres adresses ne peuvent être utilisées que sur un réseau local, ou par un accord privé de routage mutuel.
- ❖ Parmi **2000::/3** 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



4 - La couche Internet - IPV6

Blocs d'adresses IPv6

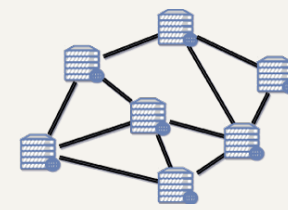
- * *Unique Local Addresses* : Les adresses **fc00::/7** sont des adresses privées, non routables sur internet (comme 10.0.0.0/8, etc. pour IPv4)
- * *Link-Local Addresses* : Les adresses **fe80::/10** sont des adresses de lien local, non routables, pour la configuration automatique
 - Les adresses **fe80::/64** 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



4 - La couche Internet - IPV6

Réservation d'adresses IPv6

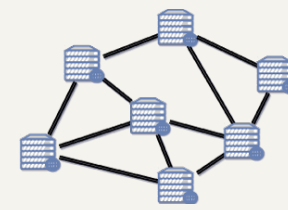
- ❖ Les adresses IPv6 (comme les adresses IPv4) sont distribuées par blocs par l'[IANA](#) (*Internet Assigned Numbers Authority*) aux RIR (*Regional Internet Registries*)
- ❖ Les RIR sont 5 :
 - [RIPE-NCC](#) pour l'Europe et le Moyen-Orient
 - ARIN : Amérique du Nord
 - APNIC : Asie & Pacifique
 - 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.)



4 - La couche Internet - IPV6

Utilisation d'IPv6

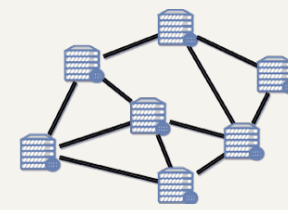
- ❖ Voir <https://6lab.cisco.com/stats/>
- ❖ [Renater](#) était le premier réseau français à utiliser IPv6
- ❖ La majorité des FAI disposent de ressources internes IPv6 et peuvent proposer des connexion IPv6 à leurs clients professionnels
 - ▶ La France est assez bien placée au niveau mondial pour le 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_1%27IPv6_par_système_d%27exploitation



4 - La couche Internet - IPV6

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.



4 - La couche Internet - IPV6

Voir aussi

- ❖ <https://networkingtoolbox.net/subnetting/ipv6-subnet-calculator>
- ❖ <https://test-ipv6.com> Test your IPv6 connectivity
- ❖ [How to Enable and Configure IPv6 on Linux: A Comprehensive Guide](#)

❖ Commandes linux

- `$ ping6 -c4 example.com`
PING6(56=40+8+8 bytes) 2a01:e0a:b30:a660:40d7:1450:9e49:b73d -->
2606:4700:10::6814:179a
16 bytes from 2606:4700:10::6814:179a, icmp_seq=0 hlim=57 time=7.664 ms
16 bytes from 2606:4700:10::6814:179a, icmp_seq=1 hlim=57 time=9.081 ms
16 bytes from 2606:4700:10::6814:179a, icmp_seq=2 hlim=57 time=8.775 ms

--- example.com ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 7.664/8.507/9.081/0.609 ms
- `ip -6 addr show`



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

Couche réseau

Client (utilisateur)

Opérateur réseau (transporteur)

Fig 3.1 - 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 vraie couche de bout en bout

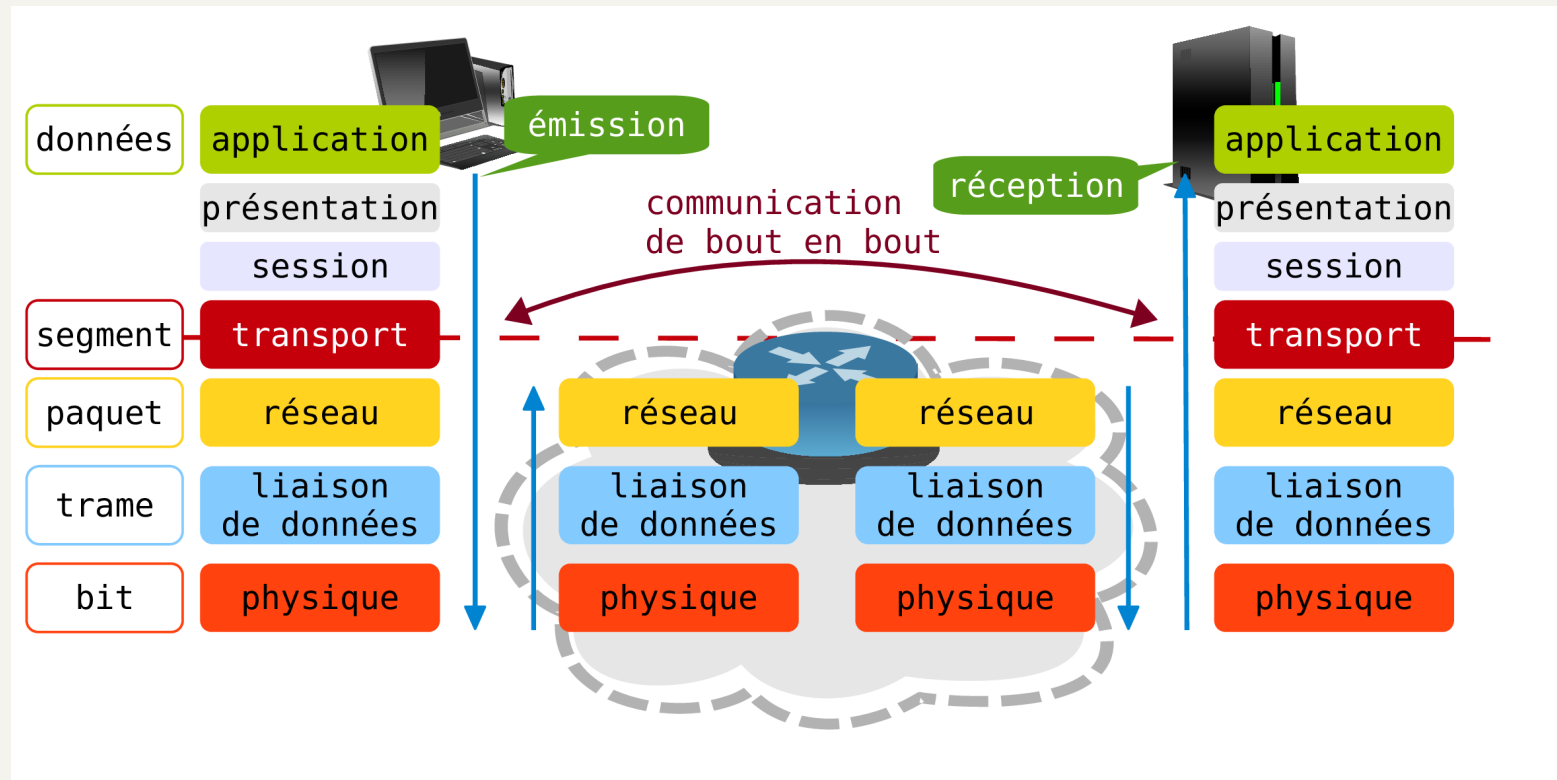


Fig 3.2 - Place de la couche transport dans le modèle OSI



3 - La couche Transport dans Internet

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*.



3 - La couche Transport dans Internet

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.



3 - La couche Transport dans Internet

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* » par laquelle une application peut envoyer et recevoir des données.





3 - La couche Transport dans Internet

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).



3 - La couche Transport dans Internet

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.



3 - La couche Transport dans Internet

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

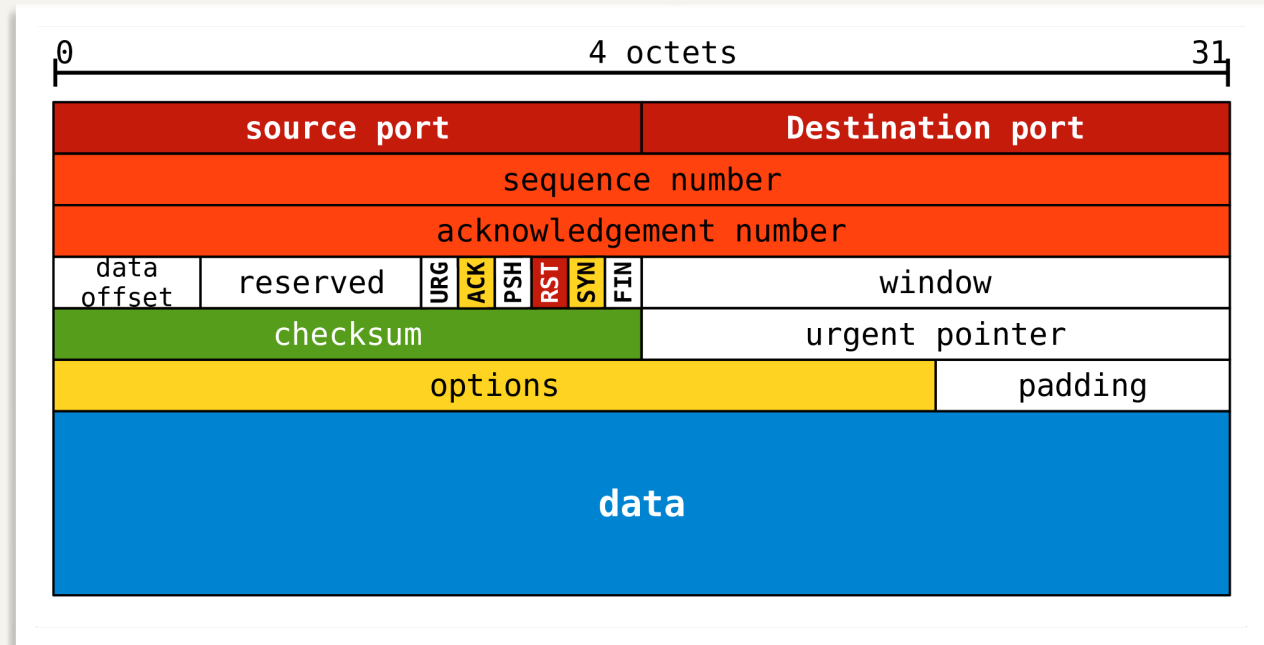


Fig 3.7 - Format d'un segment TCP



L'en-tête 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

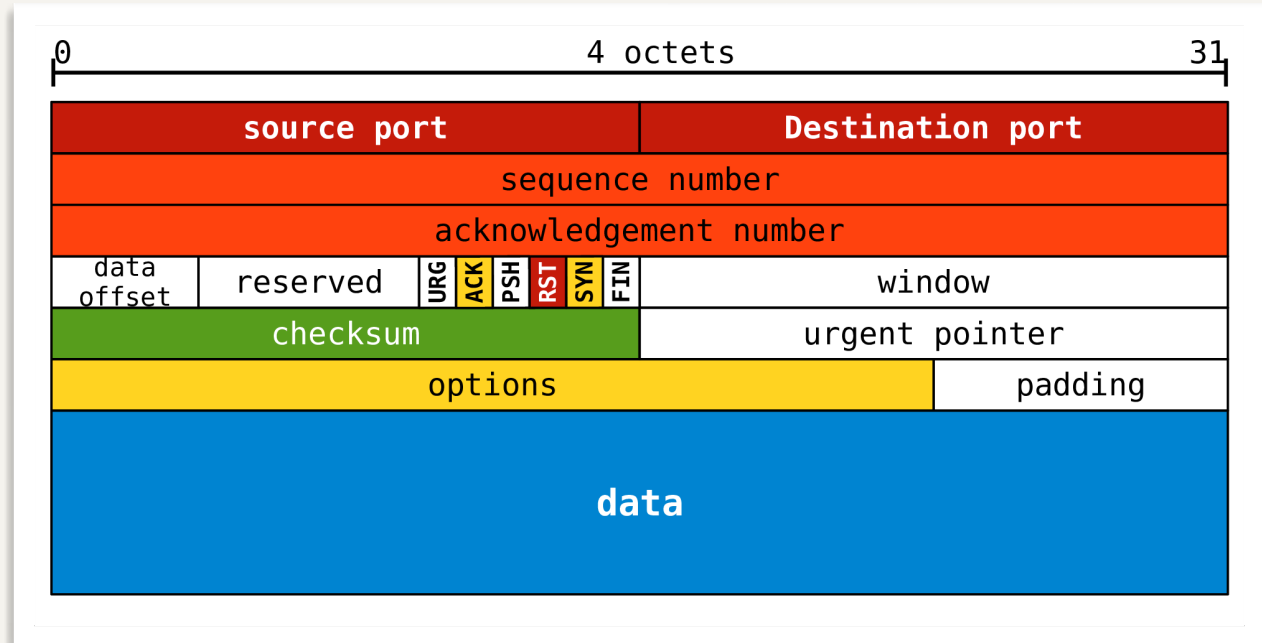


Fig 3.7 - Format d'un segment TCP



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

- ❖ *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.

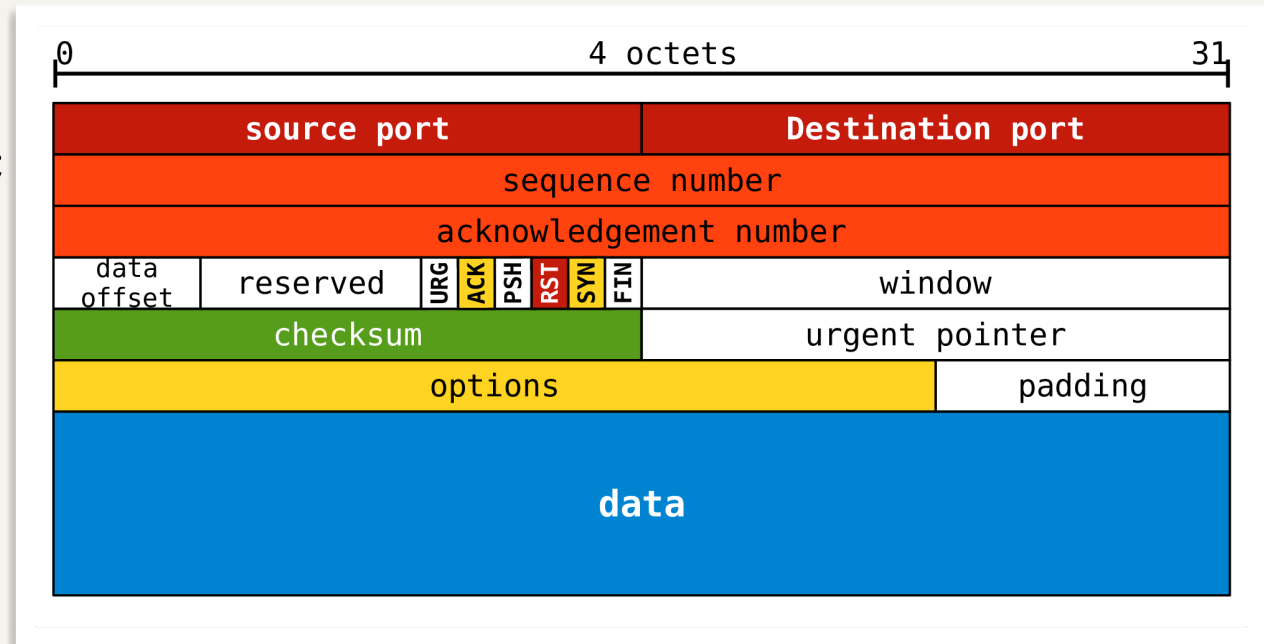


Fig 3.7 - Format d'un segment TCP



Gestion de la connexion TCP

- ❖ **Établissement de connexion** avec une poignée de main en **trois étapes** (*Three ways handshake*) : **① ② ③**
 - ▶ Avec la primitive *LISTEN*, le serveur réalise une ouverture passive d'un port
 - ▶ **①** 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**

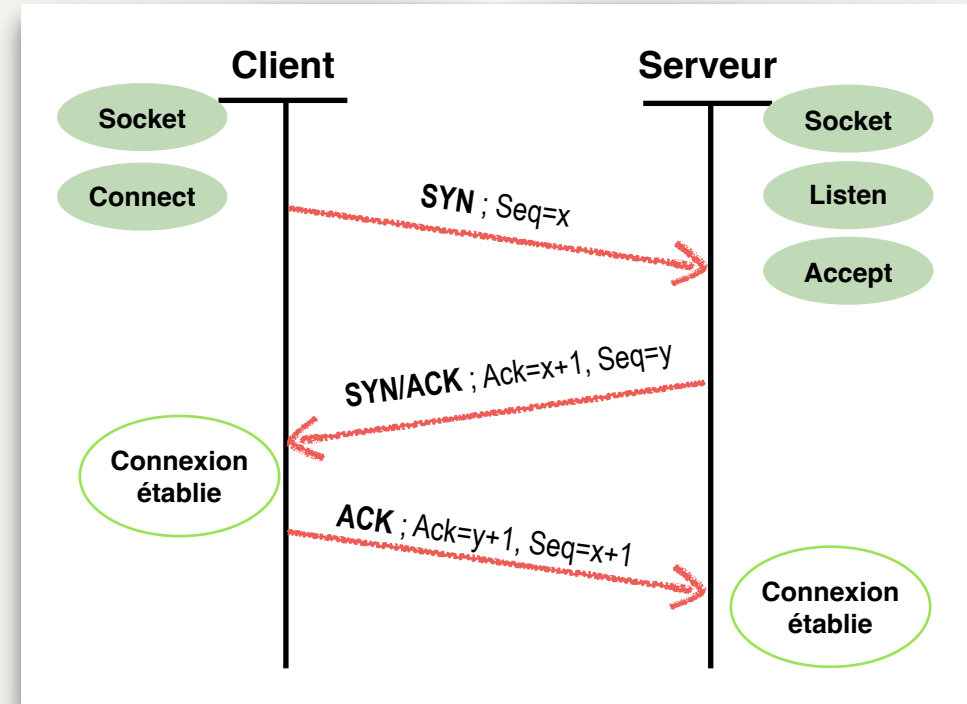


Fig 3.8 - Connexion TCP en 3 étapes



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

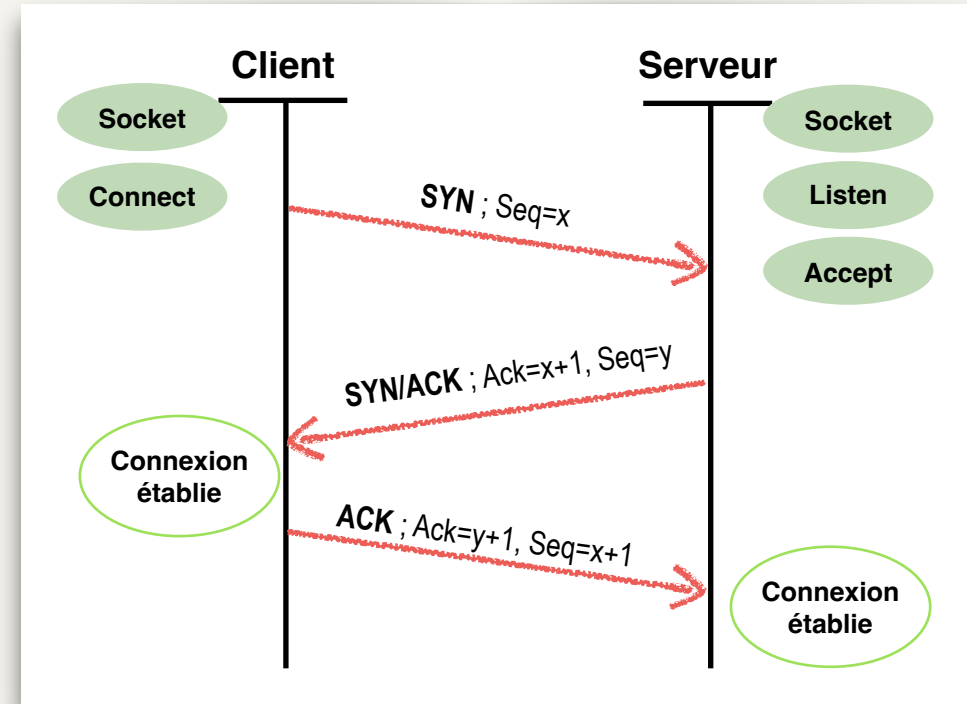


Fig 3.8 - Connexion TCP en 3 étapes

- ▶ 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

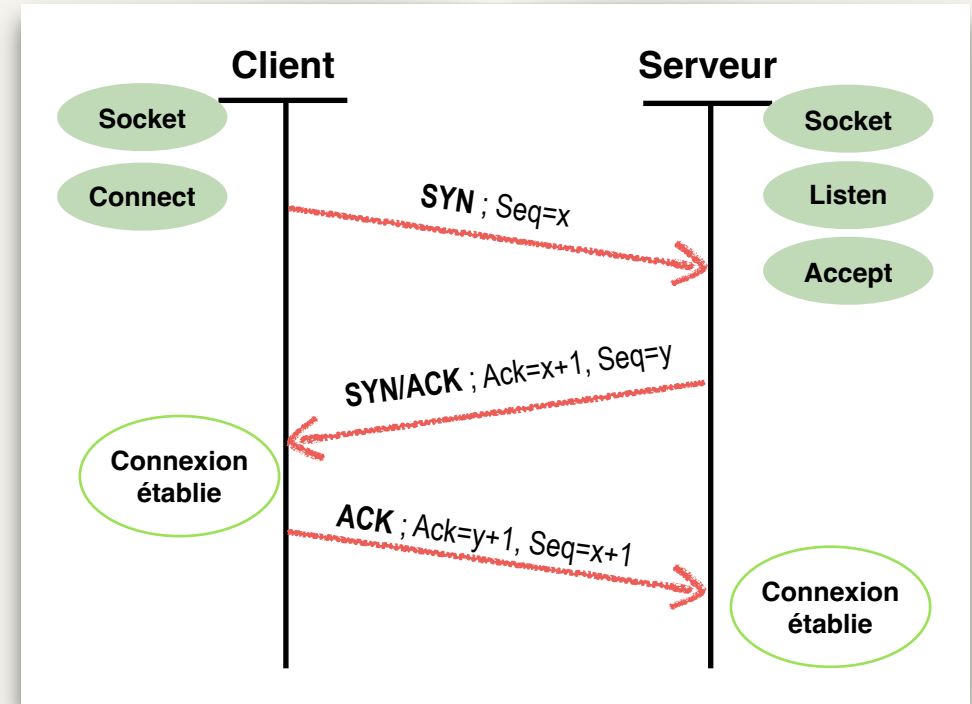


Fig 3.8 - Connexion TCP en 3 étapes

- ▶ Voir 'Comprendre les caractéristiques du TCP sous CCNA 200-301 - Vidéo Tuto d'Alphorm' : <https://www.youtube.com/watch?v=1V2B50mTGtM>



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

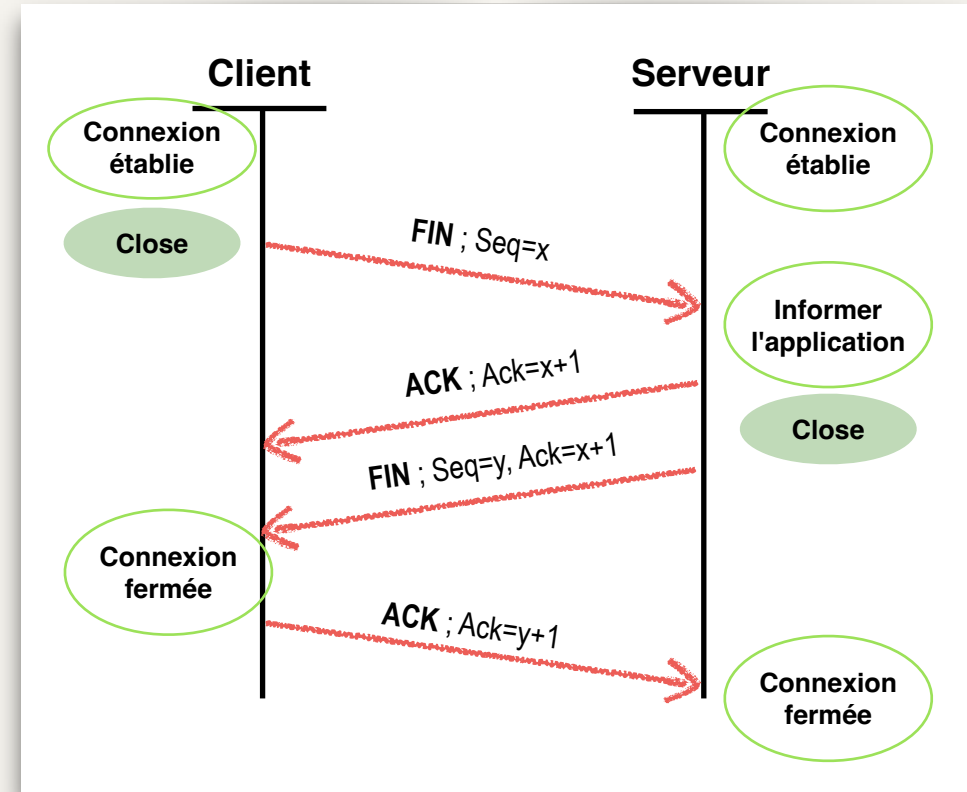


Fig 3.9 - Libération de connexion TCP



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 $N_{émis} \leq window$.



3 - La couche Transport dans Internet

À voir :

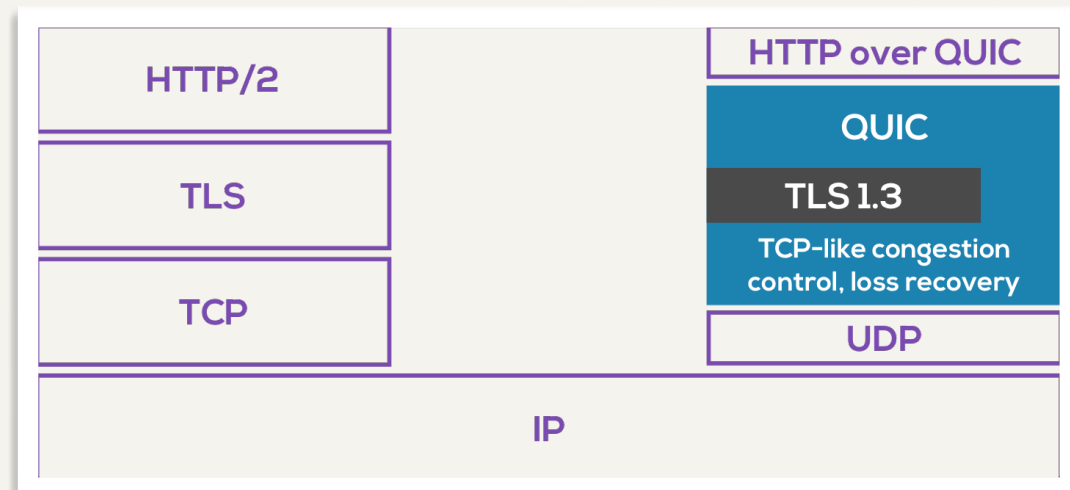
- ❖ [Comprendre les caractéristiques du TCP sous CCNA 200-301 - Vidéo Tuto](#)



3 - La couche Transport dans Internet

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

- ❖ 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

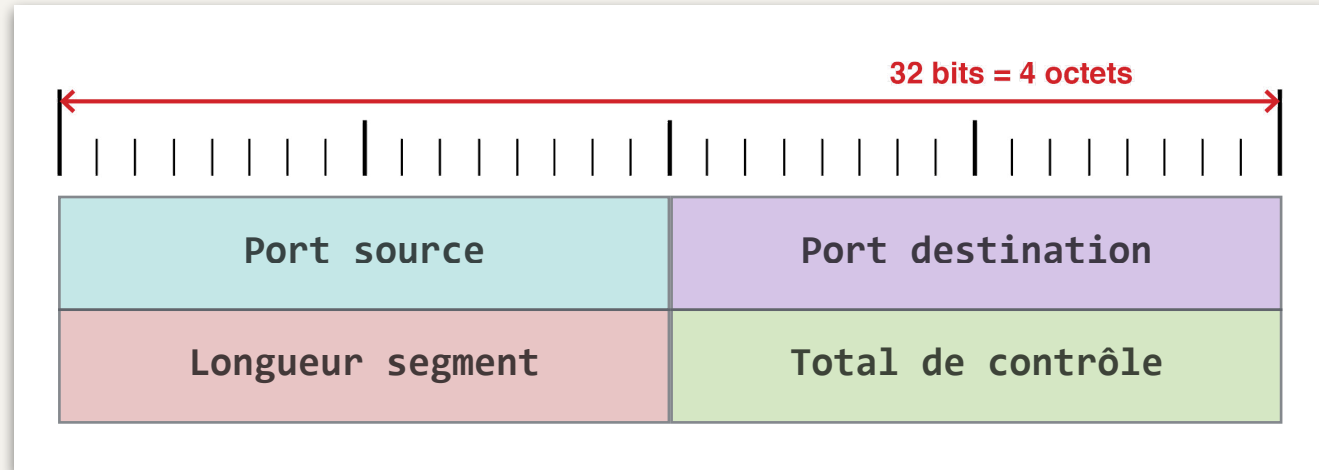


Fig 3.10 - L'en-tête UDP

1 - Rappel

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.

2 - Introduction à DNS

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...)

2 - Introduction à DNS

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.

2 - Introduction à DNS

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.

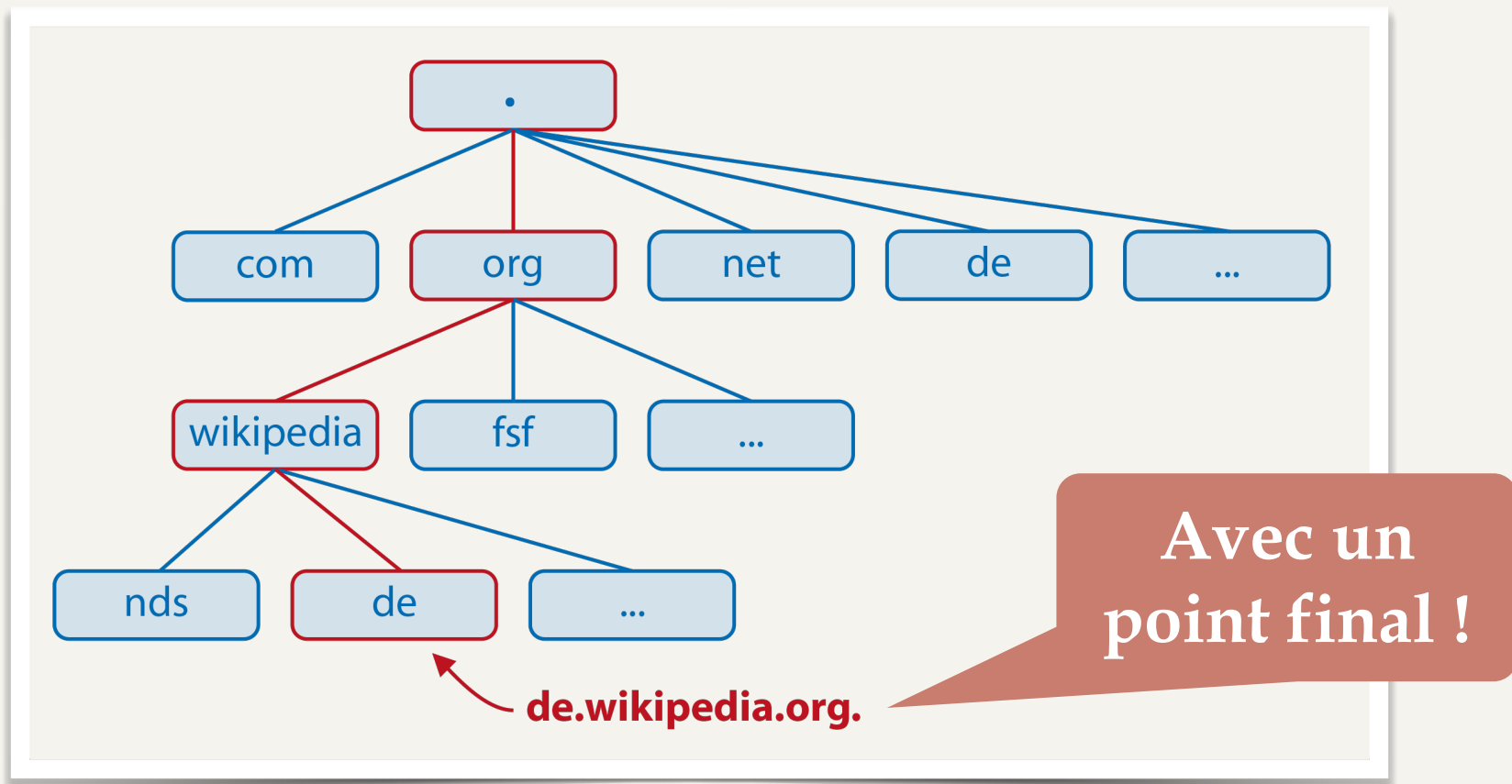


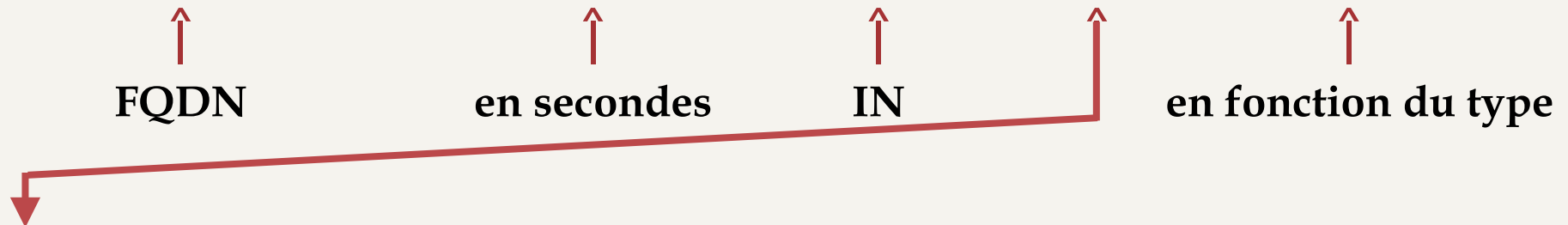
Fig 4.1 - La hiérarchie du FDQN de.wikipedia.org.

2 - Introduction à DNS

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
- ...

2 - Introduction à DNS

Enregistrement de ressources, suite

❖ Exemple :

<i>Nom de domaine</i>	<i>Durée de vie</i>	<i>Classe</i>	<i>Type</i>	<i>Valeur</i>
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.wiktictionary.org`
- `dig mx fr.wiktictionary.org`
- `host fr.wiktictionary.org`

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 4.2 - Tim Berners-Lee

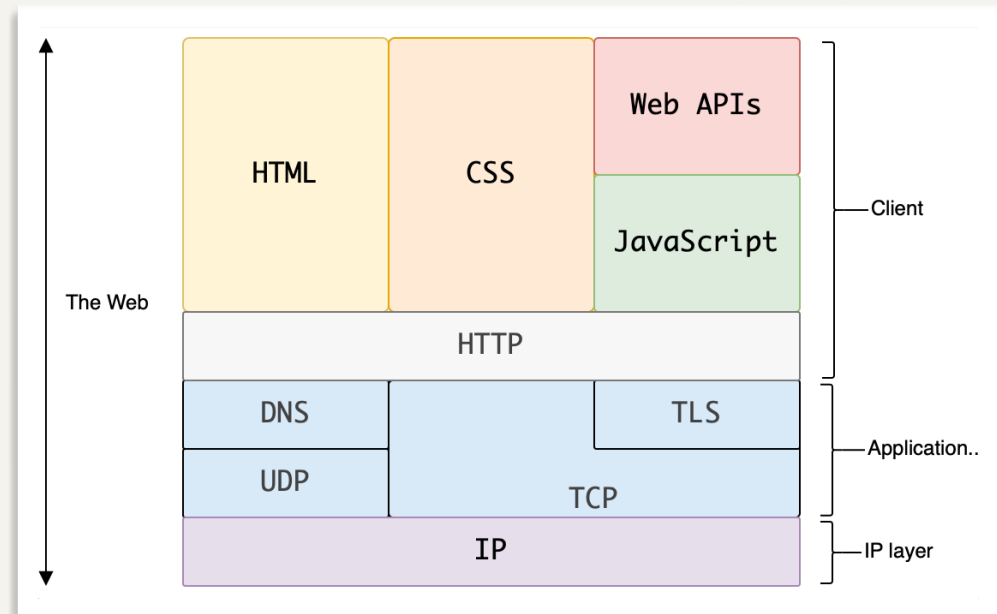


Fig 4.3 - Couches liées au web

<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.

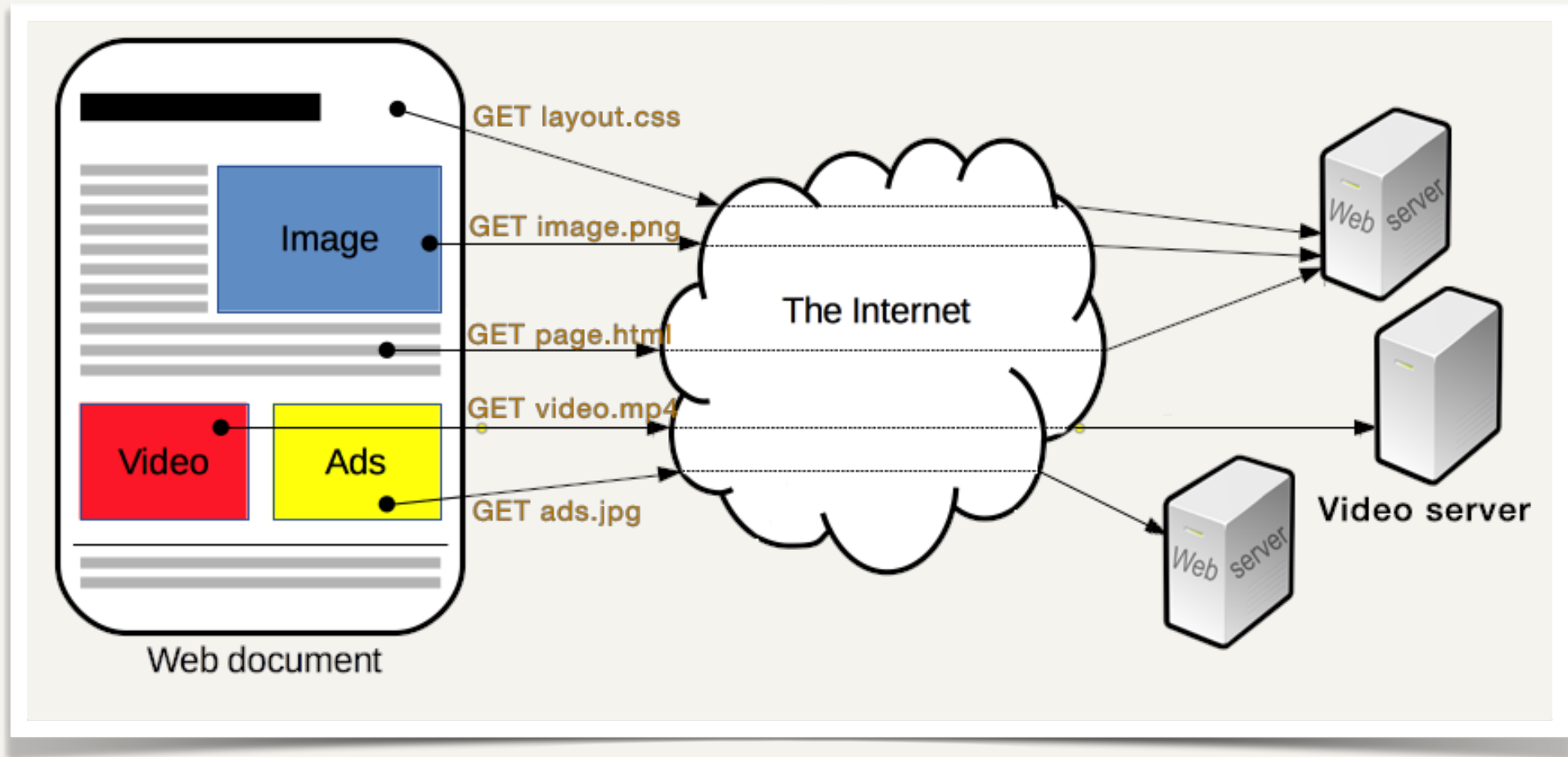


Fig 4.4 - Requêtes pour récupérer des ressources d'un document web

3 - HTTP

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, apng, jpeg, webp, svg, gif...),
 - ▶ des vidéos (mp4, webm...),
 - ▶ 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.)

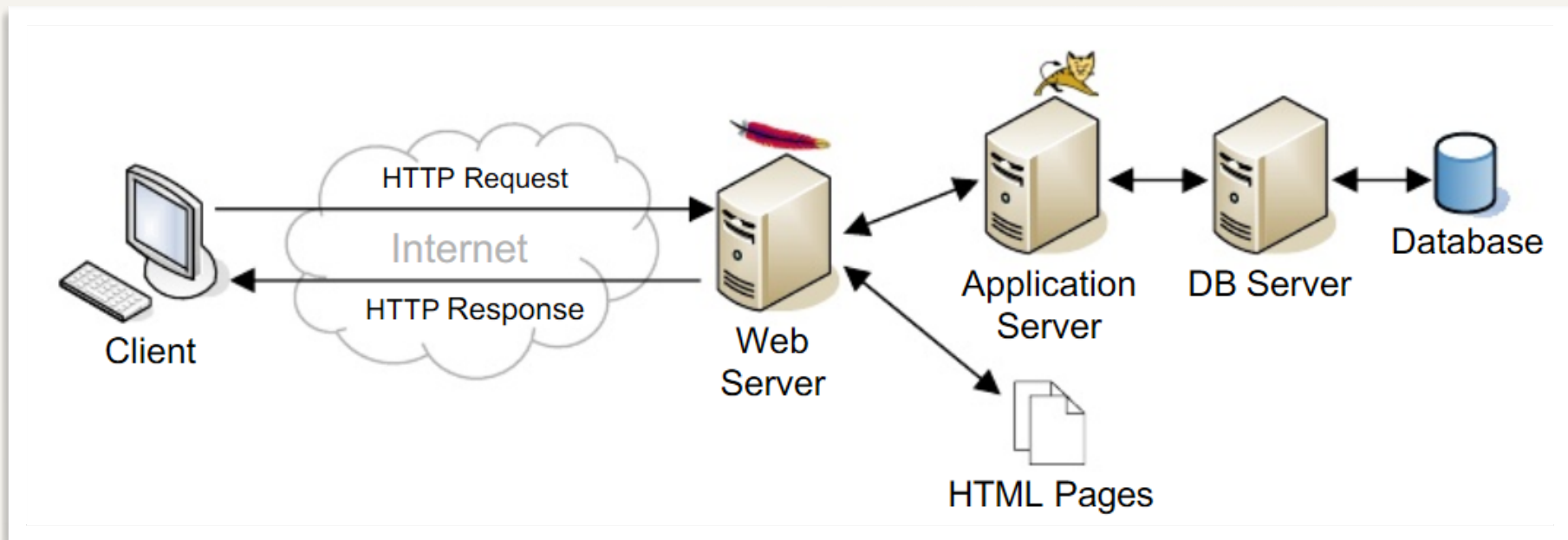


Fig 4.5 - Architecture N-Tier

3 - HTTP

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 1^{re} 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.

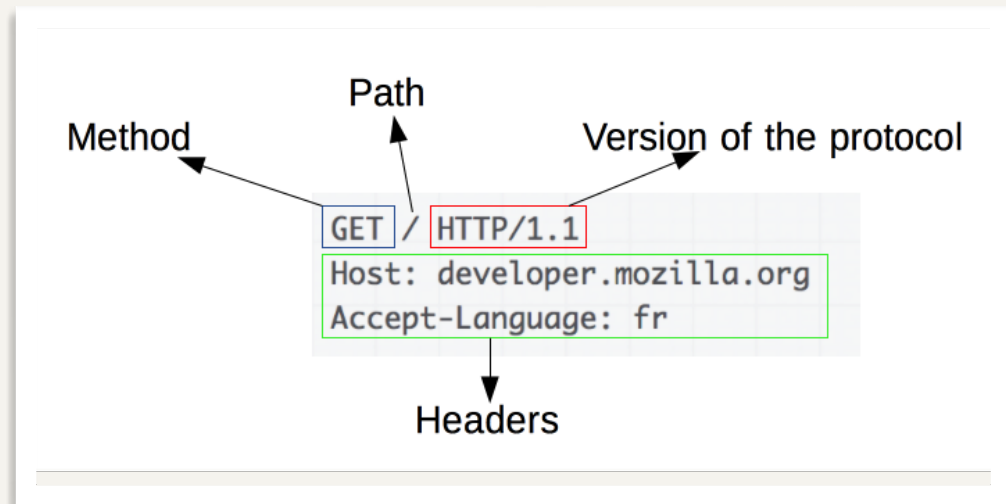


Fig 4.6 - Requête HTTP

3 - 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.

3 - HTTP

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 2^e niveau et d'un domaine de 1^{er} niveau (TLD, Top Level Domain)

3 - HTTP

Les standards du web

❖ URL, *Uniform Resource Locator*, suite...

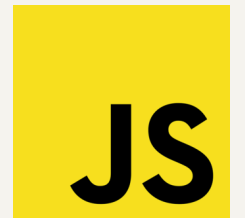
`https://archive.digital.com:8080/seq/find.php?q=url&lg=fr`



- (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 chaîne 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*).

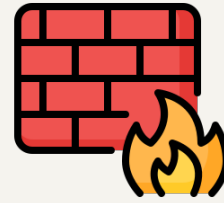


3 - HTTP

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)



1 - Préambule

Contenu du chapitre

- ❖ *Course d'obstacles en tous genres*
 - ▶ Les boitiers divers et variés encore appelés “middleboxes”
 - ▶ Etude de l'impact des équipements NAT/Firewall et protocoles associés dont DHCP