

1.1 Introduction : communiquer, mais comment ?

1.1.1 Des machines qui se parlent

C'est quoi un réseau ?

Un **réseau informatique** est un ensemble de machines reliées entre elles afin d'échanger des données. Dès 2 machines, on peut parler de réseau ! On en rencontre à toutes les échelles :

- une **salle de classe ou une maison** : quelques ordinateurs reliés à un boîtier physique,
- un **lycée ou une entreprise** : plusieurs bâtiments connectés,
- un **pays ou une région** : les infrastructures des fournisseurs d'accès à Internet (FAI),
- le **monde entier** : **Internet**, qui relie des milliards de machines via des câbles intercontinentaux.

La naissance d'Internet

Tout commence en pleine **Guerre froide**. Face à la menace nucléaire soviétique, l'armée américaine cherche un réseau de communication capable de survivre à une frappe : si un nœud est détruit, les données doivent pouvoir emprunter un autre chemin. C'est dans ce contexte qu'est créé en **1969 ARPANET**, un réseau expérimental reliant quelques universités et centres de recherche, et qui deviendra le premier ancêtre d'Internet.

Au fil des décennies, des réseaux indépendants ont émergé partout dans le monde. Plutôt que de tout fusionner, on a choisi de les **interconnecter** : **Internet est donc un réseau de réseaux**, une toile reliant des millions de réseaux autonomes qui parlent tous le même langage.

Concrètement, Internet repose en grande partie sur des **câbles physiques** : des câbles en fibre optique traversent les continents et même les fonds des océans pour relier les pays entre eux. Le Wi-Fi et les réseaux mobiles n'en sont que la partie visible et locale.

Exercice 1 — Autour de vous

1. Citez deux appareils présents dans votre lycée qui sont connectés à un réseau.

.....

2. À la maison, listez les appareils connectés à votre réseau Wi-Fi. Quels types de données s'échangent-ils ?

.....

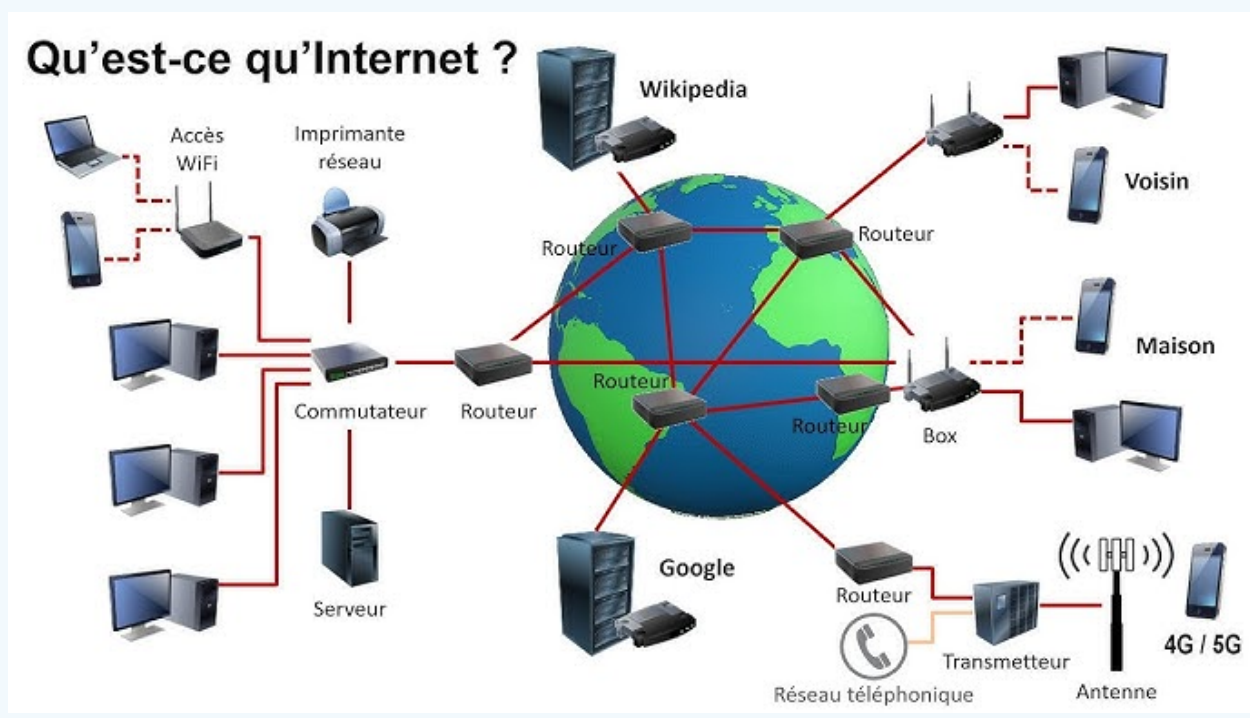
.....

1.1.2 Un réseau, des équipements

Les acteurs d'un réseau

Un réseau n'est pas seulement composé de machines utilisatrices. Pour que les données circulent correctement, plusieurs types d'équipements jouent chacun un rôle précis :

- les **machines terminales** (ordinateurs, téléphones, imprimantes...) : ce sont les **sources** et les **destinations** des données,
- les **commutateurs (switches)** : ils relient les machines d'un même réseau local entre elles,
- les **routeurs** : ils relient des réseaux différents entre eux et guident les données de réseau en réseau jusqu'à leur destination,
- les **câbles et ondes** (Ethernet, Wi-Fi, fibre optique...) : ils constituent le **support physique** par lequel transitent les données.

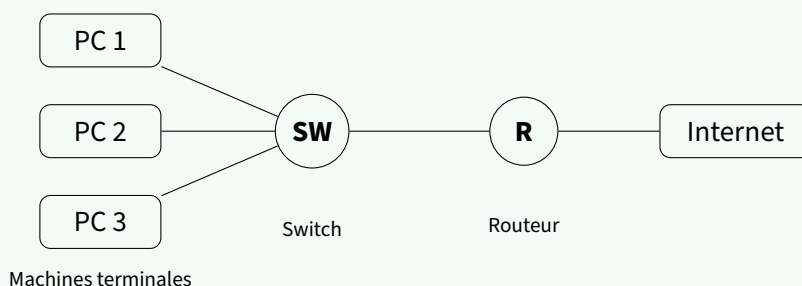


Remarque 1. A la maison, le routeur, le switch et la borne wi-fi sont un seul et même objet, la box!

Exemple 1 — Le réseau du lycée

Dans un lycée typique, on trouve :

- des ordinateurs et des tablettes dans les salles (machines terminales),
- des switches dans chaque salle, qui relient ces machines entre elles,
- un routeur principal qui connecte le réseau du lycée à Internet,
- des câbles Ethernet et des bornes Wi-Fi comme supports physiques.



1.1.3 Ce qui circule sur le réseau

Définition 1 — Données numériques

Sur un réseau informatique, tout ce qui circule est une suite de **0 et de 1**. Un texte, une image, une vidéo, un e-mail : quelle que soit sa nature, toute information est représentée et transmise sous cette forme binaire.

📄 Pourquoi découper en paquets ?

Lorsqu'une machine envoie un fichier volumineux, elle ne l'envoie pas d'un seul bloc mais le découpe en petits morceaux appelés **paquets**. Ce découpage présente plusieurs avantages :

- **Partager le réseau** : avec des paquets, plusieurs échanges peuvent s'intercaler sur le réseau plutôt qu'une seule machine le monopoliser,
- **Emprunter des chemins différents** : deux paquets d'un même fichier peuvent transiter par des routes différentes et se rejoindre à destination,
- **Gérer les erreurs** : si un paquet est perdu, il suffit de le redemander plutôt que de tout renvoyer depuis le début.

Définition 2 — Paquet

Un **paquet** est un petit morceau de données tel qu'il circule sur le réseau. Il contient une portion des données utiles ainsi que des informations d'acheminement (adresses source et destination notamment). Chaque paquet voyage **indépendamment** à travers le réseau.

1.1.4 La notion de protocole

Parler la même langue

Pour qu'un échange fonctionne, les deux parties doivent suivre les **mêmes règles**. C'est vrai dans la vie courante : deux personnes qui discutent doivent parler la même langue, respecter les tours de parole, savoir comment signaler qu'elles ont fini de parler. En informatique, ces règles communes s'appellent des **protocoles**.

Définition 3 — Protocole réseau

Un **protocole réseau** est un ensemble de règles formelles qui définissent comment deux machines échangent des données sur un réseau. Il précise notamment le **format des messages**, l'**ordre des échanges** et la **gestion des erreurs**.

Exemple 2 — Un protocole du quotidien

Lorsque vous appelez quelqu'un au téléphone, un protocole implicite s'applique :

1. l'appelant compose le numéro et attend,
2. l'appelé décroche et dit « allô »,
3. l'appelant se présente,
4. la conversation commence.

Si l'une des deux personnes ne respecte pas ce protocole (par exemple en raccrochant avant que l'autre décroche), la communication échoue.

Des protocoles, il y en a partout

Internet repose sur des dizaines de protocoles différents, chacun ayant un rôle précis :

- **IP** pour acheminer les données (*voir section 1.2*),
- **TCP** pour garantir la fiabilité d'une transmission (*voir section 1.4.1*),
- **HTTP** pour le web (*voir section 1.4.3*),
- **DNS** pour traduire les noms de domaine en adresses (*voir section 1.4.2*),
- etc...

Exercice 2 — Protocoles du quotidien

1. En dehors de l'informatique, citez un autre exemple de situation de la vie courante qui repose sur un protocole implicite. Décrivez les règles en jeu.

.....

.....

.....

.....

.....

2. Que se passe-t-il si l'une des parties ne respecte pas le protocole? Illustrez avec votre exemple.

.....

.....

.....

.....

3. Selon vous, pourquoi est-il important que les protocoles réseau soient **publics et standardisés**, plutôt que définis librement par chaque fabricant?

.....

.....

.....

.....

1.2 Une adresse pour tout le monde

On a vu qu'Internet est un réseau de réseaux, et que des routeurs guident les données de machine en machine jusqu'à destination. Mais pour qu'un paquet arrive au bon endroit, encore faut-il pouvoir **identifier précisément** chaque machine. C'est le rôle de l'**adresse IP**.

Définition 4 — Le protocole IP

IP (Internet Protocol) est le protocole fondamental d'Internet. Il définit la façon dont les paquets sont **adressés** et **acheminés** de machine en machine à travers les réseaux. C'est lui qui attribue à chaque interface réseau une adresse unique : l'**adresse IP**.

1.2.1 Les adresses IP

Définition 5 — Adresse IP

Une **adresse IP** est un identifiant numérique attribué à une **interface réseau** (ordinateur, téléphone, imprimante, etc.) utilisant le protocole IP.

Elle sert principalement à deux choses :

- **identifier** la machine destination (où le paquet doit arriver),
- permettre aux routeurs de **transférer** le paquet de réseau en réseau dans la bonne direction.

Exemple 3 — La Poste

On peut comparer une adresse IP à une adresse postale : tout comme une adresse postale permet au facteur de trouver précisément un destinataire parmi des millions, une adresse IP permet aux routeurs d'acheminer un paquet jusqu'à la bonne machine parmi des milliards.

Remarque 2. Une adresse IP n'identifie pas une personne. Elle peut changer au cours du temps (adresse attribuée par le fournisseur d'accès, changement de réseau Wi-Fi, etc.).

Définition 6 — Adresse IPv4

Une **adresse IPv4** est codée sur **32 bits**. On l'écrit sous forme **décimale pointée** :

$$a.b.c.d$$

où chaque nombre correspond à un **octet** (8 bits), donc une valeur entre 0 et 255.

i Plage de valeurs

Le plus petit nombre représentable sur 8 bits est $00000000_2 = 0$, et le plus grand est $11111111_2 = 2^7 + 2^6 + \dots + 2^0 = 255$. Cela donne 256 valeurs possibles par octet.

i IPv4, c'est trop peu

Le protocole IPv4 permet environ 2^{32} adresses distinctes, soit un peu plus de 4 milliards. Avec la généralisation des objets connectés, ce nombre s'est révélé insuffisant.

Le protocole **IPv6**, codé sur **128 bits**, a été conçu pour y remédier : il offre 2^{128} adresses possibles, un nombre considérablement plus grand. Les adresses IPv6 s'écrivent en **hexadécimal**. IPv4 et IPv6 coexistent aujourd'hui, le passage à IPv6 se faisant progressivement.

1.2.2 Conversions binaire / décimal

☰ Lecture d'un octet

Un octet $(b_7b_6b_5b_4b_3b_2b_1b_0)_2$ représente le nombre :

$$b_7 \cdot 2^7 + b_6 \cdot 2^6 + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0.$$

Chaque bit indique si l'on « prend » ou non la puissance de 2 correspondante :

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Exemple 4 — Binaire → décimal

Convertissons $(11001010)_2$ en décimal. On repère les bits à 1 et on additionne les puissances correspondantes :

$$11001010_2 = 128 + 64 + 8 + 2 = 202_{10}.$$

Définition 7 — Méthode des divisions successives

Pour convertir un entier N en binaire :

- on divise N par 2 et on note le reste (0 ou 1),
- on recommence avec le quotient,
- on lit les restes **de bas en haut**.

Exemple 5 — Décimal → binaire

Convertissons 45_{10} en binaire.

Nombre	Division par 2	Reste
45	$45 = 2 \times 22 + 1$	1
22	$22 = 2 \times 11 + 0$	0
11	$11 = 2 \times 5 + 1$	1
5	$5 = 2 \times 2 + 1$	1
2	$2 = 2 \times 1 + 0$	0
1	$1 = 2 \times 0 + 1$	1

On lit les restes de bas en haut : $45_{10} = (101101)_2$.

On vérifie : $32 + 8 + 4 + 1 = 45$. ✓

Exercice 3 — Conversions

1. Convertir en décimal :

11000000.10101000.00000001.00001110

.....

.....

.....

.....

.....

.....

.....

3. Convertir en binaire : 172_{10}

.....

.....

.....

.....

.....

.....

.....

2. Convertir en décimal :

10000010.00010000.00001010.00100001

.....

.....

.....

.....

.....

.....

.....

4. Convertir en binaire : 219_{10}

.....

.....

.....

.....

.....

.....

.....

1.2.3 Masque de sous-réseau

📖 Pourquoi un masque ?

On peut reprendre l'analogie de l'adresse postale : une adresse comme « **12 rue des Lilas, 76000 Rouen** » contient deux informations distinctes : la **ville** (qui permet d'acheminer le courrier jusqu'au bon bureau de poste) et la **rue et numéro** (qui permet de trouver la bonne maison dans cette ville).

Une adresse IP fonctionne de la même façon : elle est composée de deux parties,

- la **partie réseau** : identifie le réseau auquel appartient la machine,
- la **partie machine** : identifie la machine à l'intérieur de ce réseau.

Le **masque de sous-réseau** indique où se situe la frontière entre ces deux parties. Deux machines peuvent communiquer directement si et seulement si elles appartiennent au **même réseau**.

📖 Définition 8 — Masque de sous-réseau

Un masque de sous-réseau est un nombre de **32 bits** composé :

- de bits à 1 pour la partie **réseau**,
- de bits à 0 pour la partie **machine**.

Il est souvent noté sous la forme $/n$, où n est le nombre de bits à 1 dans le masque.

📖 Exemple 6 — Un masque

Le masque $/20$ correspond en binaire à :

$$\underbrace{11111111.11111111.1111}_{20 \text{ bits à } 1} \underbrace{0000.00000000}_{12 \text{ bits à } 0}$$

soit 255 . 255 . 240 . 0 en décimal pointé.

📖 Calcul de l'adresse réseau

L'**adresse réseau** d'une machine est obtenue en effectuant un **ET logique** entre l'adresse IP et le masque :

$$\text{adresse réseau} = \text{adresse IP} \text{ ET } \text{masque.}$$

Le ET logique bit à bit donne 1 uniquement si les deux bits valent 1, et 0 sinon. Les bits machine sont ainsi automatiquement mis à 0.

Exemple 7 — Calcul d'adresse réseau

Soit l'adresse 172.16.131.9/20.

Adresse IP : 10101100.00010000.10000011.00001001

Masque /20 : 11111111.11111111.11110000.00000000

ET logique : 10101100.00010000.10000000.00000000

L'adresse réseau est donc 172.16.128.0/20.

Exercice 4 — Déterminer l'adresse réseau

Calculer l'adresse réseau associée aux adresses suivantes :

1. 192.168.14.67/26

.....

.....

.....

.....

.....

2. 172.31.129.200/19

.....

.....

.....

.....

.....

1.2.4 Capacité d'un réseau

Définition 9 — Capacité utilisable

Si un réseau utilise n bits pour la partie réseau, il reste $32 - n$ bits pour identifier les machines, soit 2^{32-n} adresses au total.

Parmi celles-ci :

- la **première** est l'**adresse réseau** (réservée),
- la **dernière** est l'**adresse de broadcast** (réservée).

Le nombre d'**adresses utilisables** est donc $2^{32-n} - 2$.

Exemple 8 — Réseau /20

Un réseau /20 dispose de $32 - 20 = 12$ bits pour les machines, soit $2^{12} = 4096$ adresses au total.
En retirant les deux adresses réservées : $4096 - 2 = 4094$ adresses utilisables.

Exercice 5 — Capacité de réseaux

Pour chaque masque, indiquer le nombre d'adresses utilisables :

1. /16

.....
.....

2. /24

.....
.....

3. /28

.....
.....

4. /25

.....
.....

5. /30

.....
.....

1.2.5 Réseau local ou distant

Définition 10 — Réseau local

Un **réseau local** est un ensemble de machines qui partagent la même adresse réseau.

Local ou distant ?

Pour envoyer un paquet, une machine compare son adresse réseau à celle de la destination :

- si elles sont **identiques**, la communication est **locale** et le paquet est envoyé directement,
- sinon, la communication est **distante** et le paquet est confié à un **routeur par défaut**.

Exercice 6 — Communication locale ou distante ?

Indiquer si la communication est locale ou distante, en justifiant par le calcul :

1. 192.168.1.62/24 vers 192.168.1.201/24

.....

.....

.....

.....

.....

2. 172.16.15.254/20 vers 172.16.16.1/20

.....

.....

.....

.....

.....

1.3 Le modèle en couches

Quand un paquet voyage sur Internet, il ne traverse pas qu'un seul équipement. Une carte réseau, un switch, un routeur, un navigateur web... chacun intervient à un moment précis, et chacun n'a besoin de connaître que **les informations qui le concernent**. Les informaticiennes et informaticiens ont formalisé cette réalité sous la forme d'un **modèle en couches**.

1.3.1 Chaque équipement a sa mission

Des responsabilités bien séparées

Prenons l'exemple d'une requête web : on tape une adresse dans un navigateur et une page s'affiche. Derrière ce petit clic, plusieurs équipements interviennent chacun à leur niveau :

- le **navigateur** prépare ce qu'on appelle une requête HTTP (voir) : il sait ce qu'il veut demander, mais ignore tout de la façon dont ça va voyager,
- la **carte réseau** transforme les données en signaux électriques ou en ondes Wi-Fi : elle ignore complètement le contenu de la requête,
- le **switch** achemine le paquet sur le réseau local grâce aux adresses physiques des machines,
- le **routeur** guide le paquet de réseau en réseau grâce aux adresses IP.

Chaque équipement fait son travail **sans se soucier du reste**. C'est cette séparation des responsabilités que le modèle en couches cherche à capturer.

Exemple 9 — Une analogie : envoyer un colis

Lorsqu'une entreprise expédie un colis à un client, plusieurs services interviennent de façon indépendante :

1. le service commercial prépare la commande,
2. l'entrepôt emballe et étiquette le colis,
3. le transporteur choisit l'itinéraire,
4. le livreur dépose le colis à l'adresse.

Chaque service fait son travail sans se soucier du détail des autres. Le réseau fonctionne exactement de cette façon.

1.3.2 Le modèle TCP/IP

Définition 11 — Modèle TCP/IP

Le **modèle TCP/IP** est le modèle en couches utilisé sur Internet. Il organise les protocoles en **4 couches**, de la plus proche du matériel à la plus proche de l'utilisateur :

Couche	Nom	Rôle
4	Application	Protocoles utilisés par les applications (HTTP, DNS, etc.)
3	Transport	Découpage en paquets, fiabilité de la transmission (TCP, UDP)
2	Internet	Adressage et acheminement des paquets (IP)
1	Accès réseau	Transmission physique des données (Ethernet, Wi-Fi, fibre...)

Et le modèle OSI ?

Il existe un autre modèle, le **modèle OSI**, qui découpe la communication en **7 couches**. Il est plus détaillé que TCP/IP et sert souvent de référence théorique dans les cours de réseau. En pratique, c'est le modèle TCP/IP qui est utilisé sur Internet. Vous le croiserez certainement si vous poursuivez en informatique.

1.3.3 Encapsulation

Le principe de l'encapsulation

Lorsqu'un paquet est envoyé, chaque couche **ajoute ses propres informations** devant les données reçues de la couche du dessus : on appelle ces informations un **en-tête**. Ce mécanisme s'appelle l'**encapsulation**.

À la réception, chaque couche **retire l'en-tête** qui lui est destiné et transmet le reste à la couche du dessus : c'est la **décapsulation**.

Définition 12 — Trame

On appelle **trame** un paquet tel qu'il circule réellement sur le réseau : une suite de **0 et de 1** contenant, les uns après les autres, les en-têtes de chaque couche suivis des données utiles. C'est la **position** de chaque groupe de bits dans cette suite qui détermine ce qu'il signifie.

Le voyage d'une trame pour demander une page web

Émission

Application
(Navigateur)

HTTP

Prépare la requête
GET /index.html

Transport
(TCP)

TCP HTTP

Ajoute ports,
numéro de
séquence

Internet
(IP)

IP TCP HTTP

Ajoute adresses IP
src et dst

Accès réseau
(Carte réseau)

ETH IP TCP HTTP

Ajoute adresses MAC
transmet sur le
support physique

Réseau

... réseau(x) intermédiaire(s)
(routeurs, ...)

Routeur

ETH' IP TCP HTTP

Lit IP dst, décide
réécrit ETH (ETH')

Switch

ETH IP TCP HTTP

Lit MAC dst
retransmet intact

Réception

Accès réseau
(Carte réseau)

ETH IP TCP HTTP

Retire ETH
transmet vers IP

Internet
(IP)

IP TCP HTTP

Retire IP
transmet vers TCP

Transport
(TCP)

TCP HTTP

Retire TCP
transmet vers app

Application
(Navigateur)

HTTP

Reçoit GET
prépare l'en-
voi de la page

Exercice 7 — Le voyage d'une trame

On s'appuie sur le grand schéma de la page précédente.

1. Sur le schéma, la couche Accès réseau ajoute un en-tête Ethernet contenant une "adresse MAC". À votre avis, à quoi correspond cette adresse MAC ?

.....

.....

.....

2. Sur le schéma, on voit que le Routeur réécrit l'en-tête Ethernet (ETH devient ETH') à chaque saut, alors que l'en-tête IP reste intact tout au long du trajet. Pourquoi l'adresse IP ne change-t-elle pas alors que l'adresse MAC change ?

.....

.....

.....

3. On branche un câble Ethernet à la place du Wi-Fi. Quelle partie de la trame est modifiée ? Laquelle reste identique ?

.....

.....

.....

.....

4. Pourquoi le navigateur n'a-t-il pas besoin de savoir sur quel support physique les données vont voyager ?

.....

.....

.....

5. La trame qui arrive à destination est-elle identique à celle qui a quitté l'émetteur ?

.....

.....

1.4 Les protocoles fondamentaux

On sait maintenant comment les données sont organisées en couches et encapsulées dans des trames. Il est temps de regarder de plus près les protocoles qui font fonctionner Internet au quotidien.

1.4.1 TCP et UDP : fiabilité contre rapidité

Deux stratégies d'envoi

La couche transport a pour rôle d'organiser l'envoi des paquets entre deux machines. Mais selon l'application, les besoins sont très différents :

- télécharger un fichier exige que **tous les octets arrivent**, dans le bon ordre,
- regarder une vidéo en direct tolère quelques pertes, mais exige de la **rapidité**.

C'est pourquoi il existe deux protocoles de transport : **TCP** et **UDP**.

Définition 13 — TCP : Transmission Control Protocol

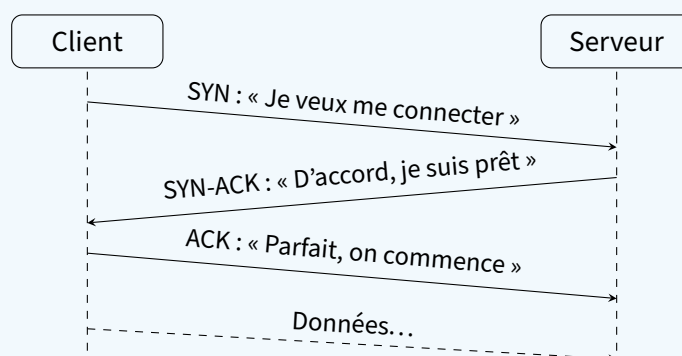
TCP est un protocole de transport **fiable** :

- il établit une **connexion** entre les deux machines avant tout envoi,
- il garantit que tous les paquets arrivent **dans le bon ordre**,
- il redemande automatiquement les paquets perdus.

En contrepartie, il est plus lent qu'UDP.

La poignée de main TCP

Avant d'envoyer des données, TCP établit une connexion en trois étapes, appelée **poignée de main (three-way handshake)** :



Ce n'est qu'une fois cette connexion établie que les données sont envoyées.

Définition 14 — UDP : User Datagram Protocol

UDP est un protocole de transport **rapide mais non fiable** :

- il envoie les paquets **sans établir de connexion** préalable,
- il ne vérifie pas que les paquets sont bien arrivés,
- les paquets perdus ne sont **pas redemandés**.

En contrepartie, il est beaucoup plus rapide que TCP.

Exercice 8 — TCP ou UDP ?

Pour chaque application, indiquer le protocole utilisé (TCP ou UDP) et justifier brièvement :

Application	Protocole	Justification
Téléchargement d'un fichier		
Chargement d'une page web		
Streaming vidéo (YouTube)		
Jeu en ligne en temps réel		
Appel vidéo (Zoom)		
Envoi d'un e-mail		

1.4.2 DNS : traduire les noms en adresses

Le problème des noms

Les machines communiquent via des adresses IP, mais les humains retiennent plus facilement `www.google.fr` que `142.250.75.196`.

Il faut donc un mécanisme pour **traduire** un nom de domaine en adresse IP. C'est le rôle du **DNS (Domain Name System)**.

Exemple 10 — L'annuaire téléphonique

Le DNS fonctionne comme un **annuaire téléphonique** : on cherche un nom (`www.google.fr`) et l'annuaire nous donne le numéro correspondant (`142.250.75.196`).

Sans DNS, il faudrait connaître l'adresse IP de chaque site que l'on visite.

Définition 15 — Résolution DNS

Lorsqu'on tape `www.google.fr` dans un navigateur, voici ce qui se passe :

1. le navigateur envoie une **requête DNS** à un serveur DNS,
2. le serveur DNS répond avec l'**adresse IP** correspondante,
3. le navigateur peut alors contacter le serveur web à cette adresse.

Cette opération s'appelle une **résolution DNS**.

Exercice 9 — DNS

1. Pourquoi Internet ne pourrait-il pas fonctionner sans DNS ?

.....
.....

2. On tape `www.lemonde.fr` dans un navigateur. Décrire dans l'ordre les étapes qui se déroulent avant que la page s'affiche, en mentionnant DNS, IP et HTTP.

.....
.....
.....
.....
.....

1.4.3 HTTP et HTTPS : le protocole du web

Définition 16 — HTTP

HTTP (HyperText Transfer Protocol) est le protocole de la couche application utilisé pour échanger des pages web. Il repose sur un modèle **requête / réponse** :

- le **client** (navigateur) envoie une **requête** au serveur,
- le **serveur** renvoie une **réponse** contenant la ressource demandée.

Structure d'une requête HTTP

Une requête HTTP suit une syntaxe précise. La première ligne contient trois éléments :

- la **méthode** : ce qu'on veut faire, GET pour demander une ressource, POST pour envoyer des données, etc.,
- le **chemin** : l'adresse de la ressource sur le serveur (ex. /index.html),
- la **version du protocole** : HTTP/1.1.

Les lignes suivantes sont des **en-têtes** qui donnent des informations supplémentaires, comme le nom du serveur (Host) ou le type de navigateur utilisé.

Comme toute donnée informatique, cette requête est en réalité une suite de 0 et de 1 sur le câble : chaque caractère est encodé selon la table ASCII puis converti en bits.

Exemple 11 — Une requête et une réponse HTTP

Lorsqu'on visite `www.example.fr/index.html`, le navigateur envoie :

```
GET /index.html HTTP/1.1
Host: www.example.fr
```

Le serveur répond :

```
HTTP/1.1 200 OK
Content-Type: text/html

(contenu de la page HTML...)
```

Le code **200** signifie que la requête a réussi. D'autres codes existent : **404** (page introuvable), **500** (erreur serveur), etc.

Définition 17 — HTTPS

HTTPS est la version **sécurisée** de HTTP. Les données échangées sont **chiffrées** grâce au protocole TLS, ce qui empêche un tiers d'intercepter ou de modifier les communications.

On reconnaît HTTPS à l'icône de cadenas dans la barre d'adresse du navigateur. Aujourd'hui, la quasi-totalité des sites web utilisent HTTPS.

⚠ Attention

HTTP transmet les données **en clair** : n'importe qui capable d'intercepter le trafic réseau peut lire les échanges, y compris les mots de passe. C'est pourquoi HTTP seul est aujourd'hui considéré comme insuffisant pour tout site manipulant des données sensibles.

Exercice 10 — HTTP et HTTPS

1. Quel est le rôle du code de statut dans une réponse HTTP? Que signifie un code 404?

.....
.....

2. On se connecte à son espace bancaire en ligne. Pourquoi est-il indispensable que le site utilise HTTPS plutôt que HTTP?

.....
.....
.....

3. En utilisant ce que l'on a vu sur le modèle en couches : à quelle couche TCP/IP appartient HTTP? Et TLS (le chiffrement de HTTPS)?

.....
.....