SNT — Internet

Partie I1 — Le binaire

Comment un ordinateur écrit un nombre, une lettre, une adresse : avec seulement des 0 et des 1

Tout ce qui circule sur Internet (un message, une photo, une vidéo, une page web) est transformé en **suites de 0 et de 1**. Avant d'étudier comment ces suites voyagent, il faut comprendre comment elles s'écrivent. C'est le thème de cette première partie.

Cours · Des 0 et des 1

Un ordinateur est fait de composants électroniques (des **transistors**) qui n'ont que **deux états** : le courant passe, ou il ne passe pas. Un processeur en contient des **milliards**. On note ces deux états **0** et **1** : c'est un **bit** (*binary digit*, « chiffre binaire »).

- Un **octet** est un groupe de **8 bits**.
- Avec **n bits**, on peut écrire $\mathbf{2^n}$ valeurs différentes (1 bit : 2 valeurs ; 2 bits : 4 ; 3 bits : 8…).
- Avec un octet, on peut donc écrire 2^8 = **256** valeurs, de 0 à 255. Tu l'as déjà rencontré : les trois nombres d'une couleur rouge–vert–bleu (thème Photographie numérique).

Cours · Compter en binaire

Dans l'écriture décimale (base 10), chaque position vaut une puissance de 10 : 155 = 1 × 100 + 5 × 10 + 5 × 1. En **binaire** (base 2), chaque position vaut une **puissance de 2**. Pour un octet :

Position	8e	7e	6e	5e	4e	3e	2e	1re
Valeur de la position	128	64	32	16	8	4	2	1
Exemple : 10011011	1	0	0	1	1	0	1	1

Du **binaire au décimal** : on additionne les valeurs des positions qui contiennent un 1.

`10011011` = 128 + 16 + 8 + 2 + 1 = **155**.

Du **décimal au binaire** : on cherche, en partant de la plus grande valeur, si elle « rentre » dans le nombre. Si oui, on écrit 1 et on la retranche ; sinon on écrit 0.

Valeur	128	64	32	16	8	4	2	1
155 − … reste	155 ≥ 128 : 1, reste 27	27 < 64 : 0	27 < 32 : 0	27 ≥ 16 : 1, reste 11	11 ≥ 8 : 1, reste 3	3 < 4 : 0	3 ≥ 2 : 1, reste 1	1 ≥ 1 : 1, reste 0

On lit les chiffres : `10011011`.

> **Attention**
>
> Un nombre binaire comme `1010` se lit « un zéro un zéro », **pas** « mille dix » : c'est le nombre dix. Quand il peut y avoir un doute, on précise la base : `1010` en base 2, `10` en base 10.

Cours · Écrire du texte : le code ASCII

Pour écrire une lettre, on lui attribue un **numéro**, puis on écrit ce numéro en binaire. Le **code ASCII** (*American Standard Code for Information Interchange*) est une table qui associe un numéro à chaque lettre non accentuée, chiffre et symbole usuel. Le code ASCII de base est écrit sur **7 bits**. L'espace a pour numéro 32. (Tu as déjà utilisé cette table dans le thème Cryptage.)

A	B	C	D	E	F	G	H	I	J	K
65	66	67	68	69	70	71	72	73	74	7

N	O	P	Q	R	S	T	U	V	W
78	79	80	81	82	83	84	85	86	87

a	b	c	d	e	f	g	h	i	j
97	98	99	100	101	102	103	104	105	106

n	o	p	q	r	s	t	u	v	w
110	111	112	113	114	115	116	117	118	119

> **Attention**
>
> La table ASCII de base **ne contient pas les lettres accentuées** (é, à, ç…). Pour écrire toutes les langues, d'autres codes plus longs ont été créés (comme l'UTF-8, utilisé par les pages web).

Tu peux t'entraîner avec le convertisseur binaire : des bits à cliquer, du texte, et même des adresses IP.

Exercices · Je compte en binaire

Exercice 1 — Marquer des oiseaux

Pour reconnaître des oiseaux, on leur accroche à la patte une **étiquette** divisée en cases. Chaque case est soit **blanche** (□), soit **coloriée en noir** (■). Deux oiseaux ne doivent jamais avoir la même étiquette.

1. Il y a **5 oiseaux** et des étiquettes de **5 cases**. Propose une étiquette pour chaque oiseau.
2. Il y a **8 oiseaux** et des étiquettes de **4 cases**. Propose une étiquette pour chaque oiseau.
3. Il y a toujours **8 oiseaux**, mais les étiquettes n'ont que **3 cases**. Est-ce possible ? Écris toutes les étiquettes.
4. Avec 2 cases, combien d'oiseaux au maximum peut-on reconnaître ? Et avec 4 cases ? Et avec n cases ?

Fichier : réponses dans ton compte rendu, exercice 1

Exercice 2 — Compter de 0 à 12

Écris les nombres de **0 à 12** en binaire.

1. Complète la liste. Aide-toi de la méthode du compteur : après `1` vient `10`, puis `11`, puis `100`…
2. Que remarques-tu sur le dernier chiffre des nombres pairs ? Et des nombres impairs ?

Réponses dans ton compte rendu (exercice 2)

Exercice 3 — Du binaire au décimal

Convertis en nombres décimaux, en détaillant les additions :

1. `10011011`
2. `1010010`
3. `1001111`

Réponses dans ton compte rendu (exercice 3)

Exercice 4 — Du décimal au binaire

Convertis en binaire, en détaillant la méthode :

1. 66
2. 78
3. 200

Réponses dans ton compte rendu (exercice 4)

Exercice 5 — Le double d'un nombre

1. Convertis en binaire les nombres 10, 15, 30, 60, puis 20, 30, 60, 120.
2. Compare l'écriture binaire d'un nombre et celle de son double (10 et 20, 15 et 30, 30 et 60, 60 et 120). Que remarques-tu ?
3. Explique cette règle avec les valeurs des positions.
4. Quelle règle simple permet de reconnaître un nombre pair écrit en binaire ?

Réponses dans ton compte rendu (exercice 5)

Exercice 6 — Combien de valeurs ? 🌶

1. Quel est le plus grand nombre que l'on peut écrire avec **un octet** ? Quel est le plus petit ?
2. Combien de nombres différents peut-on écrire avec un octet ?
3. Combien de valeurs avec **10 bits** ? Avec **16 bits** ?
4. Une couleur rouge–vert–bleu utilise 3 octets. Combien de couleurs différentes peut-on écrire ?

Réponses dans ton compte rendu (exercice 6)

Exercices · Je code du texte

Exercice 7 — Majuscules et minuscules 🌶🌶

Avec la table ASCII du cours, écris en binaire sur **7 bits** les codes des lettres : **B, A, C**, puis **b, a, c**.

1. Complète : B = 66 = `.......` ; A = …
2. Compare le code d'une majuscule et celui de la même lettre en minuscule. Que remarques-tu ?
3. Quel est l'écart entre les deux codes ? Pourquoi un seul bit change-t-il ?

Réponses dans ton compte rendu (exercice 7)

Exercice 8 — Décoder un message 🌶🌶

Voici un message écrit en code ASCII sur **7 bits** :

message.txt

```
1101111 1110101 0100000 1100101 1110011 0100000 1110100
1110101 0100000 1101110 1100101
```

1. Convertis chaque groupe de 7 bits en nombre décimal, puis en lettre avec la table ASCII. Quel est le message ?
2. Quelle lettre de ce message aurait dû porter un accent ? Pourquoi n'en voit-on pas ?
3. Réponds à la question, en écrivant ta réponse en lettres, puis en binaire ASCII sur 7 bits (sans accents, ni majuscule).

Fichier : 📝 réponses dans ton compte rendu, exercice 8

Exercices · Le binaire avec Python

Python sait convertir : `bin(155)` donne `'0b10011011'` et `int("10011011", 2)` donne `155`. Mais pour **comprendre**, on va écrire nous-mêmes les conversions.

Exercice 9 — Du décimal au binaire, en Python 🌶🌶

Recopie et exécute ce programme. Il applique la méthode du cours, de la plus grande valeur à la plus petite :

ex09.py

```
def en_binaire(n):
    texte = ""
    for puissance in [128, 64, 32, 16, 8, 4, 2, 1]:
        if n >= puissance:
            texte = texte + "1"
            n = n - puissance
        else:
            texte = texte + "0"
    return texte


print(en_binaire(155))
```

1. Quel est le résultat pour `en_binaire(155)` ? Compare avec le cours.
2. Teste avec 66, 78 puis 200. Compare avec l'exercice 4.
3. Que se passe-t-il pour `en_binaire(256)` ? Pourquoi ?
4. Modifie la fonction pour qu'elle fonctionne avec **16 bits** (jusqu'à 65 535). Teste avec 1000.

Fichier : NOM_Prenom_ex09.py

Exercice 10 — Du binaire au décimal, en Python 🌶🌶🌶

Cette fonction convertit un texte de `0` et de `1` en nombre. À chaque chiffre lu, on **double** le nombre obtenu, puis on **ajoute** le chiffre :

ex10.py

```
def en_decimal(texte):
    n = 0
    for chiffre in texte:
        n = 2 * n + int(chiffre)
    return n


print(en_decimal("10011011"))
```

1. Exécute le programme. Retrouve-t-on 155 ?
2. Complète à la main le tableau de l'évolution de n pour `"1011"` : 0, puis après le chiffre 1, puis après 0, puis après 1, puis après 1. Quel est le résultat ?
3. Pourquoi « doubler » à chaque chiffre fonctionne-t-il ? (pense à l'exercice 5)
4. Écris `est_pair(texte)` qui renvoie `True` si le nombre binaire `texte` est pair, **sans** le convertir. Teste avec `"1010010"` et `"1001111"`.

Fichier : NOM_Prenom_ex10.py

Partie I2 — Du câble à l'adresse IP

Comment est né Internet, par quels câbles passent les données, et comment chaque machine est repérée

Internet est un **réseau de réseaux** : des millions de réseaux (ceux des entreprises, des universités, des particuliers, des opérateurs) reliés entre eux. Dans cette partie, tu vois d'où vient Internet, comment les données circulent physiquement, et comment chaque machine reçoit une **adresse**.

Cours · Un peu d'histoire

Date	Événement
Années 1950–1960	Les premiers ordinateurs sont reliés à des terminaux à distance. Naît l'idée de **découper les messages en paquets**.
1969	**ArpaNet** : premier réseau à commutation de paquets, financé par l'armée américaine, qui relie quatre universités.
1971	Premier **courriel** (Ray Tomlinson), avec le symbole @. Début du projet français **Cyclades**, dirigé par **Louis Pouzin**.
1974	Vint Cerf et Bob Kahn décrivent le protocole **TCP**.
1983	ArpaNet adopte **TCP/IP** : les réseaux peuvent se connecter entre eux. C'est la **naissance d'Internet**.
1989–1991	Tim Berners-Lee invente le **Web** au CERN : pages, liens, navigateur.
Depuis 1999	Explosion des **objets connectés** : téléphones, montres, voitures, caméras…

Internet et le **Web** ne sont pas la même chose. Internet est le **réseau**, le Web est un **service** parmi d'autres qui fonctionne sur ce réseau (comme le courriel, la visioconférence ou les jeux en ligne). Cyclades a été arrêté en 1978, mais son idée centrale (les paquets sont indépendants, c'est aux machines des extrémités de vérifier que tout est arrivé) a été reprise par TCP/IP.

Cours · Par où passent les données ?

Les données voyagent sur des **liaisons** : le plus souvent un **câble** (électricité dans du cuivre, lumière dans du verre) ou des **ondes radio**. Même un message envoyé depuis un téléphone passe vite par un câble : les antennes sont reliées entre elles par de la **fibre optique**, et des câbles sous-marins relient les continents.

Liaison	Support	Débit indicatif	À savoir
Fibre optique	lumière dans un fil de verre	300 Mbit/s à 1 Gbit/s	Rapide et stable, longue distance.
ADSL	fil de cuivre du téléphone	1 à 70 Mbit/s	Le débit baisse quand on est loin du central.
Wi-Fi	ondes radio, sur quelques dizaines de mètres	dépend de la box	Partagé entre tous les appareils connectés.
4G	ondes radio, jusqu'à une antenne	30 Mbit/s environ	Partagé entre tous les utilisateurs de l'antenne.

Liaison	Support	Débit indicatif	À savoir
Satellite	ondes radio, vers l'espace	20 Mbit/s environ	Zones isolées. Le signal fait un très long trajet.

Des valeurs qui évoluent

Ces débits sont des **ordres de grandeur**. Ils changent avec la technologie, la distance, l'heure et le nombre d'utilisateurs. Le **débit réel** est souvent inférieur au débit annoncé.

Débit et latence

Deux grandeurs décrivent une liaison :

- le **débit** : la quantité de données transmises **par seconde**, en **bits par seconde** (bit/s, Kbit/s, Mbit/s, Gbit/s) ;
- la **latence** : le temps que met un signal pour aller d'un point à un autre, en **millisecondes** (ms).

Une autoroute à dix voies a un grand débit, mais le trajet reste long si on roule loin : le débit ne change pas la latence.

Attention aux unités

Le **débit** s'exprime en **bits** par seconde (b minuscule : Mbit/s ou Mb/s), la **taille d'un fichier** en **octets** (o : Mo, Go). **1 octet = 8 bits.** Une box à 100 Mbit/s transmet donc au mieux 100 ÷ 8 = **12,5 Mo** par seconde.

Pour calculer un temps de transfert : **temps (s) = taille (en bits) ÷ débit (en bit/s)**.

Les ordres de grandeur du temps

La lumière parcourt environ **300 000 km/s** dans le vide, et environ **200 000 km/s** dans une fibre optique. Aucun signal ne va plus vite : la latence ne peut pas être nulle.

Cours · Chaque machine a une adresse

Pour que des données arrivent au bon endroit, chaque machine connectée à Internet doit être repérée par une **adresse IP** (*Internet Protocol*). Dans la version utilisée depuis 1983, **IPv4**, une adresse est écrite sur **32 bits**, soit **4 octets**, présentés en décimal et séparés par des points :

adresse IPv4

```
192.0.2.42
```

Décimal	192	0	2	42
Binaire	11000000	00000000	00000010	00101010

Chaque nombre est écrit sur un octet : il est donc compris **entre 0 et 255**. Une adresse IPv4 valide est faite de **quatre nombres entre 0 et 255**, séparés par des points.

Certaines adresses sont **privées** : elles servent à l'intérieur d'un réseau local (la maison, le lycée) et ne sont pas utilisables sur Internet. Ce sont celles qui commencent par :

- `10.` (de 10.0.0.0 à 10.255.255.255) ;
- `172.16.` à `172.31.` ;
- `192.168.`.

Toutes les autres sont **publiques** : un site, un serveur ou ta box en possède une. Ta **box** fait la liaison : les appareils de la maison ont des adresses privées, et la box a **une** adresse publique pour tout le monde. Chez toi et chez ton voisin, un ordinateur peut très bien avoir la même adresse privée (`192.168.1.10`), sans risque de confusion.

Le convertisseur binaire permet de passer d'une adresse IP à ses 32 bits, et de tester si elle est valide.

Exercices · Exercices

Exercice 11 — Repères historiques 🌶

Réponds avec le tableau du cours.

1. Combien d'années séparent la création d'ArpaNet de la naissance d'Internet ?
2. Qui a dirigé le projet Cyclades, et dans quel pays ?
3. Pourquoi dit-on que 1983 est la naissance d'Internet, alors qu'ArpaNet existe depuis 1969 ?
4. Le Web et Internet sont-ils nés à la même date ? Donne un autre service d'Internet qui n'est pas le Web.

Fichier : 📝 réponses dans ton compte rendu, exercice 11

Exercice 12 — Bits et octets 🌶

Tu sais que 1 octet = 8 bits. Convertis :

1. 16 Mbit en Mo
2. 40 Mo en Mbit
3. un débit de 100 Mbit/s en Mo/s
4. un débit de 1 Gbit/s en Mo/s (1 Gbit = 1 000 Mbit)

Réponses dans ton compte rendu (exercice 12)

Exercice 13 — Temps de transfert 🌶🌶

Calcule le temps de transfert (en secondes, puis en minutes si c'est long). Pense à convertir la taille en **bits**.

1. Une photo de 4 Mo, sur une liaison à 20 Mbit/s.
2. Un fichier de 1,5 Go (1 Go = 1 000 Mo), sur une liaison à 100 Mbit/s.
3. Un film de 2 Go : en ADSL à 8 Mbit/s, puis en fibre à 500 Mbit/s.
4. Dans la vie réelle, le temps est souvent plus long que celui que tu as calculé. Donne deux raisons.

Réponses dans ton compte rendu (exercice 13)

Exercice 14 — Un câble sous l'Atlantique 🌶🌶

Le câble sous-marin **Ellalink** relie le Portugal au Brésil sur environ **6 000 km**. Quand tu envoies une demande et que tu reçois la réponse, le signal fait **l'aller et le retour**.

1. Quelle distance parcourt le signal pour un aller-retour ?
2. À 300 000 km/s (vitesse de la lumière dans le vide), combien de temps cela prend-il ? Donne le résultat en millisecondes.
3. Dans la fibre, la lumière avance à environ 200 000 km/s. Recommence le calcul.
4. Un satellite géostationnaire est à 36 000 km d'altitude. Pour un aller-retour, le signal monte et redescend **deux fois** (à l'aller puis au retour). Quelle distance totale ? Quel temps minimal ?
5. Pour une visioconférence ou un jeu en ligne, est-il préférable d'avoir un débit élevé, une latence faible, ou les deux ? Explique.

Réponses dans ton compte rendu (exercice 14)

Exercice 15 — Adresse IPv4 valide ou non ? 🌶

Pour chaque adresse, indique si elle peut être une adresse IPv4. Justifie.

1. `1.0.0.1`
2. `245.0.545.245`
3. `1.245.3.4`
4. `1.2.45.3.4`
5. `192.168.1`
6. `256.10.10.10`

Écris ensuite `192.0.2.42` en binaire, un octet après l'autre. Tu peux vérifier avec le convertisseur.

Réponses dans ton compte rendu (exercice 15)

Exercice 16 — Privée ou publique ? 🌶🌶

Indique si chaque adresse est **privée** ou **publique** :

1. `192.168.1.20`

2. `172.20.5.9`
3. `172.32.0.1`
4. `10.4.4.4`
5. `8.8.8.8`
6. `172.15.0.1`
7. Deux ordinateurs, dans deux maisons différentes, ont la même adresse `192.168.1.10`. Est-ce un problème ? Pourquoi ?

Fichier : 📝 réponses dans ton compte rendu, exercice 16

Exercice 17 — L'adresse de ma machine 🌶️

Trouve l'adresse IP de ta machine sur le réseau où tu es.

- **Windows** : ouvre l'invite de commandes et tape `ipconfig /all`.
- **Mac ou Linux** : ouvre un terminal et tape `ifconfig` (ou `ip addr`).
- **Téléphone** : réglages Wi-Fi, puis détails du réseau connecté.

1. Note l'adresse IPv4 de ta machine, la **passerelle** (c'est la box ou le routeur) et le **serveur DNS**.
2. Ces adresses sont-elles privées ou publiques ? Comment le sais-tu ?
3. Compare avec la machine d'un voisin : quels nombres sont communs ? lesquels sont différents ?

Fichier : 📝 réponses dans ton compte rendu, exercice 17

> **Ne publie pas ton adresse publique**
>
> L'adresse **publique** de ta box est une donnée personnelle : elle permet de te localiser à peu près et de relier tes visites. Ne la note pas dans un document partagé.

Exercice 18 — Assez d'adresses ? 🌶️🌶️

Une adresse IPv4 est écrite sur 32 bits.

1. Combien d'adresses IPv4 différentes existe-t-il au maximum ? (Tu peux utiliser une calculatrice : $2^{32} = 2^{10} \times 2^{10} \times 2^{10} \times 4$.)
2. La population mondiale est d'environ 8 milliards. Y a-t-il assez d'adresses IPv4 pour que **chaque habitant** en ait une ? Et si chacun a plusieurs appareils ?
3. La version **IPv6** utilise **128 bits**. Combien d'adresses cela donne-t-il (en puissance de 2) ? Est-ce plus ou moins qu'IPv4 ?
4. Comment fait-on aujourd'hui avec le manque d'adresses IPv4 ? (indice : tu as vu la réponse en lisant le cours)

Réponses dans ton compte rendu (exercice 18)

Exercice 19 — Reconnaître les adresses, en Python 🌶️🌶️🌶️

Ce programme vérifie qu'un texte est une adresse IPv4 valide. Il utilise `split` (voir le thème Les données, partie D3) pour découper le texte autour des points :

ex19.py

```
def est_ipv4(texte):
    parties = texte.split(".")
    if len(parties) != 4:
        return False
    for partie in parties:
        if not partie.isdigit():
            return False
        if int(partie) > 255:
            return False
    return True


for adresse in ["192.168.1.10", "245.0.545.245", "1.2.45.3.4",
"8.8.8.8"]:
    print(adresse, est_ipv4(adresse))
```

1. Exécute le programme. Les résultats sont-ils ceux de l'exercice 15 ?
2. Que fait la ligne `if len(parties) != 4` ? Que contient `parties` pour `"1.2.45.3.4"` ?
3. Écris une fonction `est_privee(adresse)` qui renvoie `True` pour une adresse privée (`10.`, `172.16.` à `172.31.`, `192.168.`) et `False` sinon. Teste-la avec `192.168.1.20`, `172.32.0.1` et `8.8.8.8`.

Fichier : NOM_Prenom_ex19.py

Partie I3 — Paquets, routage et TCP

Comment un message traverse le réseau : découpé en paquets, aiguillé par des routeurs, remis en ordre à l'arrivée

Quand tu envoies une photo à un ami, elle ne voyage pas d'un seul bloc le long d'un seul câble. Elle est **découpée**, chaque morceau part de son côté, puis tout est **reconstitué** à l'arrivée. Deux protocoles se partagent le travail : **IP** et **TCP**. Un **protocole** est un ensemble de règles que les machines suivent pour se comprendre.

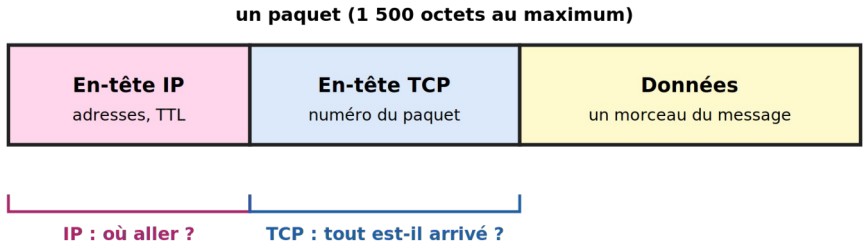
Cours · Une comparaison : envoyer un livre par la poste

Imagine que tu dois envoyer un livre de 200 pages à un ami, mais que la poste n'accepte que de **petits colis**. Tu découpes le livre en 20 paquets, tu **numérotes** chaque colis (1/20, 2/20…), tu écris l'**adresse** de ton ami et ton adresse, et tu postes tout. Les colis passent par des centres de tri : chacun choisit la route qui lui semble la meilleure, et ils n'arrivent pas forcément dans l'ordre. Ton ami les remet **dans l'ordre** grâce aux numéros. S'il en manque un, il te le **redemande**.

Dans la poste	Dans Internet	Qui s'en occupe ?
Écrire l'adresse du destinataire et de l'expéditeur	Mettre les adresses IP dans le paquet	le protocole **IP**
Aiguiller les colis dans les centres de tri	Acheminer les paquets de routeur en routeur	le protocole **IP**, par les **routeurs**
Numéroter les colis, redemander ceux qui manquent, remettre dans l'ordre	Numéroter les paquets, accuser réception, renvoyer, reconstituer	le protocole **TCP**, par les **machines** aux deux bouts

Cours · Le paquet

Le message est découpé en morceaux. Chaque morceau est enveloppé dans un **paquet**, qui contient trois choses :

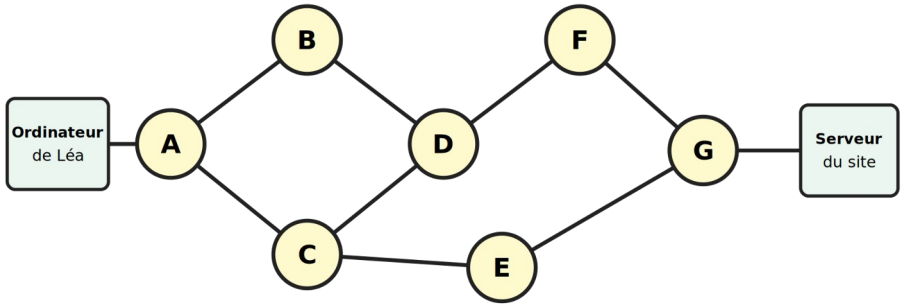


Un paquet : deux en-têtes (comme l'enveloppe) et un morceau du message.

- l'**en-tête IP** : l'adresse IP de l'**émetteur**, celle du **destinataire**, et une durée de vie (le **TTL**, voir plus bas) ;
- l'**en-tête TCP** : le **numéro** du paquet dans le message, pour pouvoir remettre les morceaux dans l'ordre ;
- les **données** : le morceau du message. Un paquet transporte au plus environ 1 500 octets.

Cours · Le routage

Les paquets traversent des **routeurs**. Un routeur est une machine qui a plusieurs liaisons : quand un paquet arrive, il lit l'adresse du **destinataire** et décide vers quel routeur voisin l'envoyer. C'est le **routage**.



Chaque rond est un routeur ; chaque trait est une liaison.

Un petit réseau : le paquet part de l'ordinateur de Léa, passe par plusieurs routeurs et arrive au serveur.

- Chaque routeur ne connaît que ses **voisins** et une **table de routage** qui lui indique « pour aller vers telle adresse, passe par tel voisin ». Il ne connaît pas le chemin complet.
- Le chemin **n'est pas décidé à l'avance** : chaque routeur choisit au moment où le paquet arrive.
- Deux paquets d'un même message peuvent donc suivre des **chemins différents**.
- Si un routeur ou une liaison tombe en panne, les paquets suivants passent **par un autre chemin**, s'il en existe un : c'est ce qui rend Internet **robuste**.

Pour éviter qu'un paquet **tourne en rond** dans le réseau à cause d'une erreur de routage, chaque paquet porte un compteur, le **TTL** (*time to live*, « durée de vie »). **Chaque routeur retire 1** au TTL. Quand le TTL tombe à **0**, le routeur **détruit** le paquet. Un paquet est donc destiné à traverser **un nombre maximal de routeurs**.

Cours · TCP : un transport fiable

Le routage ne garantit rien : un paquet peut être **perdu** (un routeur saturé le détruit), **retardé**, ou arriver **dans le désordre**. C'est **TCP** (*Transmission Control Protocol*) qui répare cela, aux deux extrémités :

1. à l'envoi, TCP **découpe** le message et **numérote** les paquets ;
2. à l'arrivée, TCP envoie un **accusé de réception** pour chaque paquet reçu ;
3. si un accusé ne revient pas, l'émetteur **renvoie** le paquet ;
4. le destinataire **remet les morceaux dans l'ordre** et reconstitue le message.

IP et TCP, deux rôles différents

IP : adresses et acheminement, **sans garantie** de livraison. **TCP** : numérotation, accusés, renvoi et remise en ordre, pour que le message arrive **complet et dans l'ordre**.

On parle de « **TCP/IP** » parce que les deux fonctionnent ensemble.

Limites du routage

Internet fonctionne en **« mieux possible »** : aucun paquet n'a la garantie d'arriver vite, ni même d'arriver. Quand le réseau est très utilisé, les routeurs sont saturés, les paquets attendent ou sont perdus, et la communication ralentit. De plus, le contenu d'un paquet est **lisible** par les routeurs qu'il traverse, sauf s'il a été **chiffré** (voir le thème Cryptage).

Pour voir tout cela à l'œuvre, utilise le simulateur de routage : tu envoies un message, tu fais tomber des routeurs, tu perds des paquets, tu actives ou non TCP.

Exercices · Exercices

Exercice 20 — IP ou TCP ?

Pour chaque action, indique si elle est réalisée par **IP** ou par **TCP**, et si elle l'est par les **routeurs** ou par les **machines des extrémités** :

1. écrire l'adresse du destinataire dans le paquet ;
2. numéroter les paquets du message ;
3. choisir le routeur suivant pour un paquet ;
4. redemander un paquet qui n'est pas arrivé ;
5. remettre les paquets dans l'ordre à l'arrivée.

Fichier : réponses dans ton compte rendu, exercice 20

Exercice 21 — Combien de paquets ?

Un paquet transporte au plus **1 500 octets** de données (1 Mo = 1 000 000 octets).

1. Une photo de 3 Mo : combien de paquets faut-il ?
2. Un message de 32 caractères (1 caractère = 1 octet) découpé en morceaux de 4 octets : combien de paquets ?
3. Un film de 2 Go (2 000 000 000 octets) : combien de paquets ? (arrondis au paquet supérieur)
4. Pourquoi ne pas envoyer le film en un seul gros paquet ? Donne une raison.

Réponses dans ton compte rendu (exercice 21)

Exercice 22 — Remettre dans l'ordre

Un ordinateur reçoit les paquets d'un même message, dans cet ordre d'arrivée. Chaque ligne donne le **numéro** du paquet (en-tête TCP) et son **contenu**.

Ordre d'arrivée	Numéro	Contenu
1er	5	`di d`
2e	2	`ez-v`

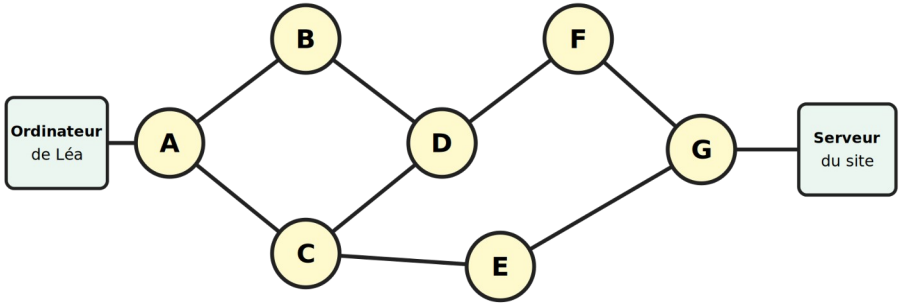
Ordre d'arrivée	Numéro	Contenu
3e	8	`CDI`
4e	1	`Rend`
5e	4	`a mi`
6e	3	`ous`
7e	7	`t le`
8e	6	`evan`

1. Quel est le message ? (un espace compte comme un caractère, il est parfois au début ou à la fin d'un contenu)
2. Le paquet n° 6 n'est jamais arrivé. Que fait TCP ?
3. Combien de paquets contiendrait le message si on avait découpé en morceaux de **8** caractères au lieu de 4 ?

Réponses dans ton compte rendu (exercice 22)

Exercice 23 — Choisir un chemin 🌶🌶

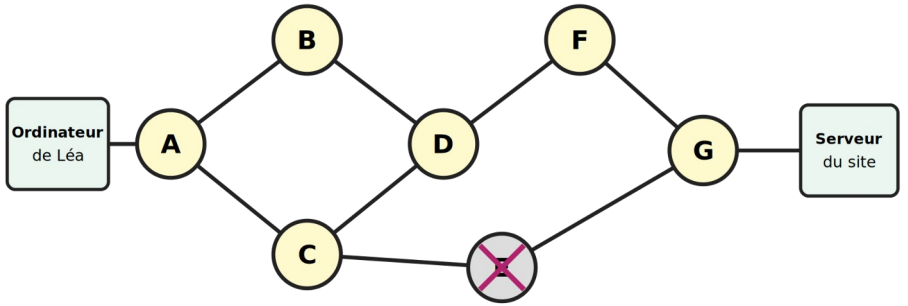
Sur ce réseau, un paquet doit aller du **routeur A** (le plus proche de l'ordinateur de Léa) au **routeur G** (le plus proche du serveur).



Chaque rond est un routeur ; chaque trait est une liaison.

Le réseau de l'exercice.

1. Quel est le chemin le plus court en nombre de routeurs ? Combien de routeurs le paquet traverse-t-il ?
2. Le routeur E tombe en panne (figure suivante). Donne un nouveau chemin possible et le nombre de routeurs traversés.
3. Combien de chemins différents existe-t-il entre A et G, sans passer deux fois par le même routeur ? Liste-les.
4. Pourquoi, en pratique, deux paquets d'un même message peuvent-ils ne pas suivre le même chemin ?



Chaque rond est un routeur ; chaque trait est une liaison.

Le routeur E est en panne (barré).

Fichier : réponses dans ton compte rendu, exercice 23

Exercice 24 — La durée de vie d'un paquet 🌶🌶

Chaque routeur qui reçoit un paquet **retire 1** à son TTL. Si le TTL tombe à **0**, le routeur détruit le paquet.

1. Un paquet part avec un TTL de 3 et suit le chemin A – C – E – G. Écris le TTL après chaque routeur. Le paquet arrive-t-il ?
2. Quel TTL minimum faut-il pour que le paquet arrive à destination par ce chemin de 4 routeurs ?
3. Les ordinateurs envoient en général des paquets avec un TTL de 64 ou de 128. Pourquoi est-ce suffisant ?
4. Une erreur de routage fait tourner un paquet en boucle entre quatre routeurs. Que se passe-t-il avec un TTL de 64 ? Que se passerait-il s'il n'y avait pas de TTL ?

Réponses dans ton compte rendu (exercice 24)

Exercice 25 — Le chemin d'un paquet 🌶🌶

La commande `tracert` (Windows) ou `traceroute` (Mac, Linux) affiche les routeurs traversés pour atteindre une machine. Voici le résultat d'un essai, avec des adresses d'exemple :

tracert 192.0.2.42

```
1    <1 ms    <1 ms    <1 ms  192.168.1.1
2     8 ms     7 ms     8 ms  203.0.113.1
3    12 ms    11 ms    12 ms  198.51.100.9
4    25 ms    24 ms    26 ms  198.51.100.77
5     *        *        *     Délai d'attente de la demande
dépassé.
6    27 ms    26 ms    27 ms  192.0.2.42
```

Chaque ligne est un **saut** : le numéro, trois temps de réponse (trois essais), puis l'adresse de la machine. Une étoile * signifie qu'il n'y a **pas eu de réponse**.

1. Combien de sauts y a-t-il jusqu'à la destination `192.0.2.42` ?
2. La première ligne est `192.168.1.1` : de quelle machine s'agit-il ? Comment le sais-tu ?
3. À la ligne 5, que signifient les trois étoiles ? Le trajet a-t-il échoué ?
4. Entre quelles lignes le temps de réponse augmente-t-il le plus ? Que peut-on en conclure sur la liaison ?
5. `tracert` fonctionne en envoyant des paquets de TTL 1, puis 2, puis 3… Explique pourquoi cette méthode permet de découvrir les routeurs un par un.

Réponses dans ton compte rendu (exercice 25)

Exercice 26 — TCP avec ou sans ? 🌶🌶

Ouvre le simulateur de routage. Envoie le message proposé, avec une **perte de 10 %** des paquets.

1. **Sans TCP** : le message reçu est-il complet ? Dans l'ordre ? Recommence trois fois et note ce que tu observes.
2. **Avec TCP** : le message reçu est-il complet ? Combien de **tours** (envois) faut-il ?
3. Mets la perte à **0 %**, puis fais tomber un routeur. Les paquets arrivent-ils ? Par où passent-ils ?
4. Résume en une phrase le rôle de TCP.

Fichier : 📝 réponses dans ton compte rendu, exercice 26

Exercice 27 — Vrai ou faux ? 🌶🌶

Pour chaque affirmation, dis si elle est vraie ou fausse. Si elle est fausse, corrige-la.

1. Le chemin d'un paquet est décidé par l'ordinateur qui l'envoie, avant le départ.
2. Deux paquets d'un même message suivent toujours le même chemin.
3. Le protocole IP garantit que tous les paquets arrivent.
4. Si un routeur tombe en panne, Internet peut continuer à fonctionner.
5. Les paquets arrivent toujours dans l'ordre où ils ont été envoyés.
6. Un routeur connaît le chemin complet jusqu'à la destination.
7. Le contenu d'un paquet est automatiquement chiffré pendant son voyage.
8. Plus le réseau est utilisé, plus les paquets risquent d'être retardés ou perdus.

Fichier : 📝 réponses dans ton compte rendu, exercice 27

Exercice 28 — Découper et reconstituer, en Python 🌶🌶🌶

Ce programme découpe un message en paquets numérotés. Les paquets sont rangés dans un **dictionnaire** (voir le thème Les données, partie D3) : la **clé** est le numéro, la **valeur** est le morceau.

ex28.py

```
def decouper(message, taille):
    paquets = {}
    numero = 1
    for debut in range(0, len(message), taille):
        paquets[numero] = message[debut:debut + taille]
        numero = numero + 1
    return paquets
```

```
message = "Rendez-vous a midi devant le CDI"
paquets = decouper(message, 4)
print(paquets)
```

1. Exécute le programme. Compare avec l'exercice 22.
2. Remplace 4 par 5 : combien de paquets obtiens-tu ? Que contient le dernier ?
3. Écris une fonction `reconstituer(paquets, total)` qui rebâtit le message : pour chaque numéro de 1 à `total` (le nombre de paquets annoncé), ajoute le morceau à un texte. Vérifie que tu retrouves le message avec `total = 8`.
4. Supprime un paquet avec `del paquets[6]`, puis modifie `reconstituer` : si un numéro est absent (`if numero not in paquets`), elle affiche `Paquet manquant` et le numéro. C'est le rôle de TCP.

Fichier : NOM_Prenom_ex28.py

Partie I4 — Les noms de domaine (DNS)

Comment « www.musee-pixel.fr » devient une adresse IP : l'annuaire d'Internet

Tu tapes `www.musee-pixel.fr` dans ton navigateur, mais les paquets ont besoin d'une **adresse IP** pour être acheminés. Quelqu'un doit donc traduire le nom en adresse. C'est le travail du **DNS**.

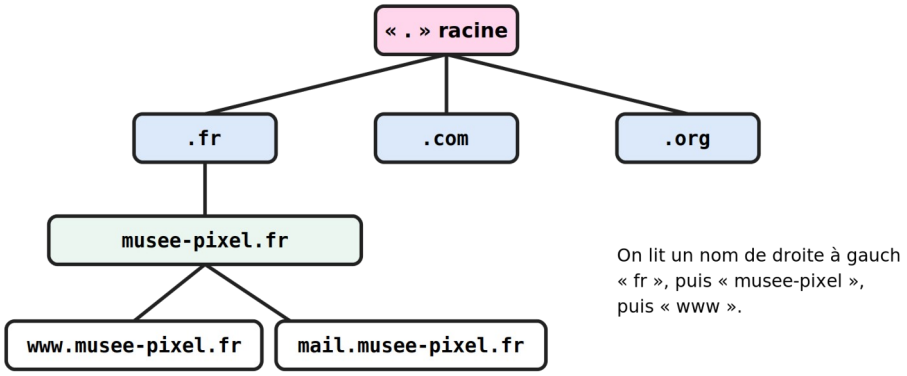
Cours · Un nom plutôt qu'un numéro

Un humain retient mal `203.0.113.10` et retient bien `www.musee-pixel.fr`. On appelle ce nom une **adresse symbolique**. Elle se lit **de droite à gauche**, du plus général au plus précis :

Partie	Rôle	Exemple
.fr	le **domaine de premier niveau** (pays ou type de site : .fr, .com, .org)	`.fr`
musee-pixel	le **nom de domaine** choisi par l'organisation (il est réservé auprès d'un bureau d'enregistrement)	`musee-pixel.fr`
www	un **sous-domaine** : une machine ou un service de l'organisation	`www.musee-pixel.fr`, `mail.musee-pixel.fr`

Cours · Le DNS : un annuaire réparti

Le **DNS** (*Domain Name System*, « système de noms de domaine ») est un **annuaire** qui associe un nom à une adresse IP. Il n'est pas rangé dans un seul ordinateur, mais **réparti sur des milliers de serveurs**, organisés en hiérarchie :



La hiérarchie du DNS : la racine connaît les serveurs de chaque terminaison, chaque terminaison connaît les domaines.

Voici ce qui se passe quand tu demandes `www.musee-pixel.fr` :

1. ton navigateur demande l'adresse au **serveur DNS** de ton réseau (celui de ta box ou de ton opérateur), appelé **résolveur** ;
2. si le résolveur ne la connaît pas, il interroge un **serveur racine**, qui lui indique les serveurs du **.fr** ;
3. le serveur du **.fr** lui indique le serveur de **musee-pixel.fr** ;
4. ce dernier connaît l'adresse de www et la donne ;
5. le résolveur renvoie l'adresse IP au navigateur, qui peut alors contacter le serveur web.

Le cache

Le résolveur **mémorise** (garde en **cache**) les réponses un certain temps. Si tu demandes de nouveau le même site, ou si un camarade du même réseau le demande, la réponse est immédiate : on ne refait pas toute la chaîne.

Un point sensible

Si le DNS ne fonctionne plus, **les sites existent toujours**, mais on ne les trouve plus par leur nom : on obtient « site introuvable ». Le serveur DNS voit aussi **tous les noms de sites** que tu demandes : le choix du résolveur est important pour la vie privée.

Pour voir une résolution étape par étape, utilise le résolveur DNS.

Exercices · Exercices

Pour les exercices, voici l'annuaire d'un petit réseau (les adresses sont des adresses d'exemple réservées à la documentation) :

Nom	Adresse IP
www.musee-pixel.fr	203.0.113.10
mail.musee-pixel.fr	203.0.113.25
www.lycee-exemple.fr	198.51.100.20
cours.lycee-exemple.fr	198.51.100.21
www.exemple.com	192.0.2.80
wiki.exemple.org	192.0.2.99

Exercice 29 — Lire une adresse symbolique

1. Pour `cours.lycee-exemple.fr` : donne le domaine de premier niveau, le nom de domaine et le sous-domaine.
2. Même question pour `wiki.exemple.org`.
3. Quelle est l'adresse IP de `mail.musee-pixel.fr` ? Et de `www.exemple.com` ?
4. Quel nom correspond à l'adresse `198.51.100.20` ?
5. Pourquoi utilise-t-on des noms plutôt que des adresses IP ? Donne deux raisons.

Réponses dans ton compte rendu (exercice 29)

Exercice 30 — Dans quel ordre ?

Voici six étapes, dans le désordre, pour afficher `www.musee-pixel.fr` :

- **a)** le navigateur envoie sa demande de page à l'adresse IP obtenue ;
- **b)** le résolveur interroge le serveur racine, puis le serveur du `.fr`, puis celui de `musee-pixel.fr` ;
- **c)** l'utilisateur tape `www.musee-pixel.fr` ;
- **d)** le navigateur demande l'adresse IP au serveur DNS du réseau (le résolveur) ;
- **e)** le résolveur renvoie l'adresse IP au navigateur ;
- **f)** le serveur web envoie la page.

1. Remets les étapes dans l'ordre.
2. À quelle étape le cache peut-il faire gagner du temps ?

Réponses dans ton compte rendu (exercice 30)

Exercice 31 — Qui interroge qui ?

Observe la figure de la hiérarchie du DNS dans le cours.

1. Pour résoudre `wiki.exemple.org`, quels serveurs sont interrogés, dans l'ordre ?
2. Pourquoi n'y a-t-il pas **un seul serveur** qui connaîtrait toutes les adresses d'Internet ? Donne deux raisons.
3. Combien de serveurs le résolveur interroge-t-il si la réponse est déjà dans son cache ?

Réponses dans ton compte rendu (exercice 31)

Exercice 32 — Utiliser le résolveur

Ouvre le résolveur DNS.

1. Demande l'adresse de `www.lycee-exemple.fr`. Combien d'étapes (serveurs interrogés) ? Note l'adresse obtenue.
2. Redemande la même adresse, avec le cache activé. Qu'observes-tu ?

3. Tape un nom qui n'existe pas (par exemple `www.musee-pixel.com`). Quel message obtiens-tu ?
4. Utilise la recherche inverse : quel nom correspond à `192.0.2.99` ?

Fichier : 📝 réponses dans ton compte rendu, exercice 32

Exercice 33 — Lire le résultat de nslookup 🌶🌶

La commande `nslookup` (Windows, Mac, Linux) interroge le DNS. Voici un résultat :

nslookup

```
> nslookup www.musee-pixel.fr
Serveur :   box.home
Address:    192.168.1.1

Réponse ne faisant pas autorité :
Nom :       www.musee-pixel.fr
Address:    203.0.113.10
```

1. Quelle est l'adresse IP de `www.musee-pixel.fr` ?
2. Quel serveur DNS a répondu ? Cette adresse est-elle privée ou publique ? Qu'en déduis-tu ?
3. La phrase « Réponse ne faisant pas autorité » signifie que la réponse vient de la **mémoire du résolveur**, et non du serveur du domaine. De quel mécanisme s'agit-il ?
4. Si tu as le droit, essaie `nslookup wikipedia.org` sur ta machine. Quelle adresse obtiens-tu ? Est-elle privée ou publique ?

Réponses dans ton compte rendu (exercice 33)

Exercice 34 — Où est la panne ? 🌶🌶

Dans chaque situation, dis si le problème vient plutôt du **DNS** ou de la **connexion** :

1. Léa tape `www.exemple.com` : « site introuvable ». En tapant l'adresse `192.0.2.80`, la page s'affiche.
2. Plus aucune page ne s'affiche, même en tapant des adresses IP.
3. Le site `www.musee-pixel.fr` déménage chez un autre hébergeur : son adresse IP change. Les visiteurs doivent-ils changer quelque chose ?
4. Pourquoi le choix du serveur DNS touche-t-il à la vie privée ?

Réponses dans ton compte rendu (exercice 34)

Exercice 35 — Un annuaire en Python 🌶🌶

Un dictionnaire convient très bien pour un annuaire : la **clé** est le nom, la **valeur** est l'adresse IP.

ex35.py

```python
annuaire = {
    "www.musee-pixel.fr": "203.0.113.10",
    "mail.musee-pixel.fr": "203.0.113.25",
    "www.lycee-exemple.fr": "198.51.100.20",
    "cours.lycee-exemple.fr": "198.51.100.21",
    "www.exemple.com": "192.0.2.80",
    "wiki.exemple.org": "192.0.2.99",
}


def resoudre(nom):
    if nom in annuaire:
        return annuaire[nom]
    return "introuvable"


print(resoudre("www.exemple.com"))
print(resoudre("www.exemple.fr"))
```

1. Exécute le programme. Que renvoie `resoudre` pour un nom absent ?
2. Écris `domaine(nom)` qui renvoie les deux dernières parties du nom (par exemple `"lycee-exemple.fr"` pour `"cours.lycee-exemple.fr"`). Utilise `split(".")` et `parties[-2]`, `parties[-1]`.
3. À l'aide de `domaine`, écris un programme qui affiche **tous les noms** et adresses du domaine `musee-pixel.fr`.

Fichier : NOM_Prenom_ex35.py

Partie I5 — Pair à pair, impacts et affiche

Deux manières de partager des fichiers, qui contrôle Internet, ce qui le menace, et une affiche pour en parler

Dans cette dernière partie, tu étudies une autre façon de faire circuler des fichiers (le **pair à pair**), tu te demandes **à qui appartient Internet** et **ce qui peut mal tourner**, puis tu réalises une **affiche**.

Cours · Client–serveur ou pair à pair

Dans le modèle **client–serveur** (celui du Web, voir le thème Le Web), un **serveur** détient les fichiers et des **clients** les lui demandent. Quand beaucoup de clients demandent le même fichier, le serveur est **saturé** : il doit envoyer autant de copies qu'il y a de clients.

Dans le modèle **pair à pair** (*peer to peer*, ou **P2P**), les machines sont **toutes à la fois clientes et serveuses** : chaque ordinateur (un **pair**) télécharge des morceaux du fichier chez d'autres pairs, et **partage** à son tour ceux qu'il a déjà reçus. Il n'y a pas de serveur central (ou très peu).

	Client–serveur	Pair à pair
Qui détient le fichier ?	le serveur	tous les pairs, par morceaux
Quand il y a beaucoup de demandes	le serveur est saturé	plus il y a de pairs, plus ça va vite
Si un élément tombe en panne	le serveur tombe : plus de fichier	les autres pairs continuent
Coût	le propriétaire paie le serveur et le trafic	réparti entre les participants
Contrôle	centralisé, facile à contrôler	dispersé, difficile à contrôler

Usages légitimes : distribuer des logiciels libres ou leurs mises à jour, partager des œuvres sous licence libre, des services de téléphonie ou de calcul partagé. **Usages illicites** : partager une œuvre protégée (film, musique, jeu) **sans l'autorisation** de ses auteurs. C'est du **piratage**, puni par la loi. En France, l'**Arcom** surveille ces pratiques et peut envoyer des avertissements.

> **Les risques du pair à pair**
>
> Ton **adresse IP** est visible par les autres pairs ; les fichiers téléchargés peuvent contenir des **virus** ; tu peux, sans le savoir, **partager** des fichiers que tu n'as pas le droit de diffuser.

Cours · À qui appartient Internet ?

À personne : chaque réseau appartient à celui qui l'a construit (opérateurs, entreprises, universités, États), et tous acceptent d'utiliser les mêmes protocoles. Mais certaines choses sont **coordonnées** au niveau mondial : les adresses IP et les noms de domaine (organismes comme l'ICANN), et les normes (IETF).

Internet est aussi **matériel** : des câbles (sous-marins ou terrestres), des antennes, des **centres de données**, des routeurs, qui consomment de l'énergie et appartiennent à des entreprises. Et il n'est pas accessible à tous de la même manière : on parle de **fracture numérique**.

La **neutralité du Net** est le principe selon lequel les opérateurs doivent traiter **tous les paquets de la même façon**, sans ralentir ni favoriser un site, un service ou un utilisateur.

Estimer le trafic. Une vidéo en qualité courante demande un débit de l'ordre de 3 Mbit/s. En 1 heure (3 600 s), cela représente 3 × 3 600 = 10 800 Mbit, soit **1 350 Mo**, donc environ **1,35 Go**. Multiplié par des millions de spectateurs, on mesure l'importance du trafic.

Cours · Ce qui menace Internet

Menace	En quoi ça consiste	Se protéger
Hameçonnage (*phishing*)	un faux message (courriel, SMS) imite une organisation pour te faire donner un mot de passe ou des données	vérifier l'adresse de l'expéditeur et du lien, ne jamais donner de mot de passe par message
Rançongiciel	un logiciel malveillant	sauvegardes régulières

Menace	En quoi ça consiste	Se protéger
	chiffre tes fichiers et réclame de l'argent pour les rendre	(règle 3-2-1, thème Les données), mises à jour
Déni de service (DDoS)	des milliers de machines infectées envoient en même temps trop de demandes à un serveur, qui est submergé	difficile pour un particulier ; ne pas laisser son ordinateur être infecté
Écoute d'un réseau	quelqu'un lit les paquets qui passent (Wi-Fi public)	https, chiffrement (thèmes Le Web et Cryptage), éviter les Wi-Fi ouverts

- Cybermalveillance.gouv.fr : conseils et assistance en cas de piratage : https://www.cybermalveillance.gouv.fr/
- CNIL : protéger ses données personnelles : https://www.cnil.fr/

Exercices · Exercices

Exercice 36 — Mille téléchargements 🌶🌶

Un éditeur de logiciels propose un fichier de **1 Go**. **1 000 personnes** le téléchargent en même temps.

1. En client–serveur, quel volume de données le serveur doit-il envoyer en tout ? (1 To = 1 000 Go)
2. Sa liaison a un débit de 1 Gbit/s. Combien de temps (en secondes, puis en heures et minutes) faut-il pour tout envoyer ?
3. En pair à pair, les téléchargeurs s'échangent les morceaux : si le serveur n'envoie qu'**une fois** chaque morceau, quel volume envoie-t-il ? Par combien est-il divisé ?
4. Pourquoi le pair à pair a-t-il un intérêt pour distribuer des logiciels libres ?

Réponses dans ton compte rendu (exercice 36)

Exercice 37 — Légal ou pas ? 🌶

Pour chaque situation, indique si l'usage du pair à pair est **légitime** ou **illicite**, et pourquoi :

1. Télécharger une distribution du système Linux avec un logiciel de pair à pair.
2. Télécharger un film sorti la semaine dernière, sans l'acheter ni le louer.
3. Recevoir une mise à jour d'un jeu vidéo, que l'éditeur propose en pair à pair.
4. Partager avec ta famille des photos de vacances que tu as prises.
5. Partager l'album d'un chanteur que tu as acheté sur CD, avec des inconnus.
6. Partager une vidéo sous licence libre, dont l'auteur autorise la diffusion.

Fichier : réponses dans ton compte rendu, exercice 37

Exercice 38 — Estimer un trafic 🌶🌶

Une vidéo en qualité courante demande un débit d'environ **3 Mbit/s**.

1. Quel volume de données (en Go) correspond à **1 heure** de vidéo ?
2. Dans ta classe, 30 élèves regardent chacun 1 heure de vidéo. Quel volume en tout ?
3. Quelqu'un regarde en moyenne 1 heure de vidéo par jour. Quel volume cela fait-il en un an (365 jours) ?
4. Donne deux moyens de réduire ce trafic.

Réponses dans ton compte rendu (exercice 38)

Exercice 39 — La neutralité du Net 🌶🌶

La **neutralité du Net** signifie que les opérateurs doivent traiter tous les paquets de la même façon.

1. Explique avec tes mots ce que cela veut dire, avec un exemple.
2. Donne **un argument en faveur** de la neutralité du Net.
3. Certains opérateurs disent qu'il faut parfois **gérer** le trafic (par exemple, donner la priorité à une visioconférence médicale sur un téléchargement). Donne un argument qui peut aller **contre** une neutralité absolue.
4. Quelle est ta position, et pourquoi ?

Fichier : réponses dans ton compte rendu, exercice 39

Exercice 40 — Menace et parade

Associe chaque situation à une menace du cours (hameçonnage, rançongiciel, déni de service, écoute d'un réseau) et à une protection :

1. Tu reçois un SMS : « Votre colis est bloqué, cliquez pour payer 1,99 € de frais », avec un lien étrange.
2. Tous tes documents sont devenus illisibles ; un message te demande de l'argent.
3. Le site de ton lycée devient inaccessible : des milliers de machines lui envoient des demandes en même temps.
4. Dans un café, quelqu'un lit tes échanges sur un Wi-Fi ouvert.

Réponses dans ton compte rendu (exercice 40)

Exercice 41 — Mini-projet : une affiche

Réalise, **seul ou à deux**, une **affiche au format A3** qui explique un sujet de ce thème à des élèves de ton âge. Choisis **un** sujet :

- **le trajet d'un message** : paquets, routeurs, TCP ;
- **les adresses** : adresse IP, noms de domaine, DNS ;
- **le pair à pair** : principe, intérêts, usages licites et illicites ;
- **protéger ses données sur Internet** : hameçonnage, rançongiciel, bonnes pratiques.

Exigences :

- un **titre** clair, un **schéma** que tu as dessiné (ou une figure du cours que tu cites), et **au moins une définition** avec tes mots ;
- au moins **deux sources vérifiées** (site officiel, encyclopédie, cours), citées en bas de l'affiche ;
- une affiche **lisible** : texte assez gros, contraste suffisant (teste-le avec le testeur de contraste du thème Le Web), pas d'information transmise uniquement par la couleur ;
- des informations **exactes** : relis ton cours avant de rendre.

Dépose ton travail dans Pronote, au format `.odt`, `.docx` ou `.pdf`. Critères d'évaluation (sans note chiffrée) :

Critère	Ce qu'on regarde
Exactitude	les informations sont justes et vérifiées
Clarté	on comprend le sujet sans explication orale
Lisibilité	taille du texte, contraste, mise en page aérée
Sources	au moins deux sources fiables, citées
Travail personnel	schéma et définitions écrits avec ses mots

- Cybermalveillance.gouv.fr : https://www.cybermalveillance.gouv.fr/
- CNIL : https://www.cnil.fr/

Fichier : NOM_Prenom_affiche.odt (ou .docx, .pdf)