

# Chapitre 7. Les réseaux

## Table des matières

### [1. Communication](#)

### [2. Eléments de base](#)

- [2.1 Les protocoles](#)
- [2.2 Les protocoles TCP et IP](#)

### [3. Rôle des différents protocoles de communication](#)

- [3.1 Le service Système de Noms de Domaine](#)
- [3.2 Récupération de paquets](#)
- [3.3 Adresse IP](#)

### [4. Architecture d'un réseau](#)

- [4.1 Différentes méthodes de connexion](#)
- [4.2 Quelques lignes de commande Linux](#)
  - [4.2.1 ifconfig](#)
  - [4.2.2 ping](#)
  - [4.2.3 traceroute](#)
  - [4.2.4 getent hosts](#)

Remplissez le jupyter notebook suivant en vous aidant de votre [livre de Première NSI de Serge BAYS](#) .

- Pour répondre, double-cliquez sur **Réponse** et complétez la zone en-dessous. Puis cliquez sur le bouton *Exécuter*.
- **Important : pour fermer votre jupyter notebook, cliquez sur :**

*Fichier / Créer une nouvelle sauvegarde*

puis sur :

*Fichier / Fermer et Arrêter*

- Ecrivez ci-dessous votre prénom et votre nom :

Réponse :

# Chapitre 7. Les réseaux

## 1. Communication

Lisez le paragraphe **Communication** p. 191  
puis répondez aux questions.

\*1) Quel est l'un des principaux intérêts d'appareils comme les ordinateurs, les tablettes et les smartphones ?\*

Réponse :

\*2) De quel réseau militaire descend le réseau mondial Internet ?\*

Réponse :

\*3) De quand date l'idée de découper un message en segments indépendants qui connaissent chacun l'adresse du destinataire ?\*

Réponse :

\*4) Quel est l'avantage de découper et d'envoyer en segments un gros message, plutôt que d'envoyer le gros message en une seule fois ?\*

Réponse :

\*5) Quel est l'inconvénient d'un réseau organisé en étoile ?\*

Réponse :

\*6) De quelles années date le protocole TCP/IP ?\*

Réponse :

\*7) Qui sont les inventeurs du protocole TCP/IP ?\*

Réponse :

## **2. Eléments de base**

### **2.1 Les protocoles**

## Introduction

Visualisez la vidéo suivante qui explique la notion générale de **protocole**.

Pour activer la sortie audio sur la prise jack de votre Raspberry ouvrez un terminal puis :

- \$ **sudo raspi-config**
- Sélectionner *Advanced Options* avec les flèches de direction puis appuyez sur la touche Entrée.
- Sélectionner *Audio* avec les flèches de direction puis appuyez sur la touche Entrée.
- Sélectionner *Force 3.5mm ('headphone') jack* avec les flèches de direction.
- Appuyez sur la touche Tabulation pour surligner *OK*. Appuyez sur la touche Entrée.
- Appuyez sur la touche Tabulation pour surligner *Finish*. Appuyez sur la touche Entrée.

Branchez vos écouteurs dans la prise jack de votre Raspberry.

Vous pouvez régler le volume en cliquant sur l'icône Son en haut à droite de l'écran.

Lisez la vidéo en cliquant sur l'image ci-dessous :

Qu'est-ce qu'un protocole ?

- Protocole Humain.

Alice

Superman



[http://www.astrovirtuel.fr/jupyter/19\\_pnsi\\_cours/protocole.mp4](http://www.astrovirtuel.fr/jupyter/19_pnsi_cours/protocole.mp4)

Lisez le paragraphe **Transmission de données** p. 191 et première moitié de la p. 192 puis complétez :

\*8) Qu'est-ce qu'un protocole ?\*

Réponse :

\*9) Que signifient les lettres **\*\*HTTP\*\*** ?\*

Réponse :

\*10) Que signifient les lettres **TCP** ? Attention ! Il y a une erreur dans le livre sur la signification du T. Vous corrigerez donc cette erreur en allant faire une recherche sur Internet.\*

Réponse :

\*11) Que signifient les lettres **IP** ?\*

Réponse :

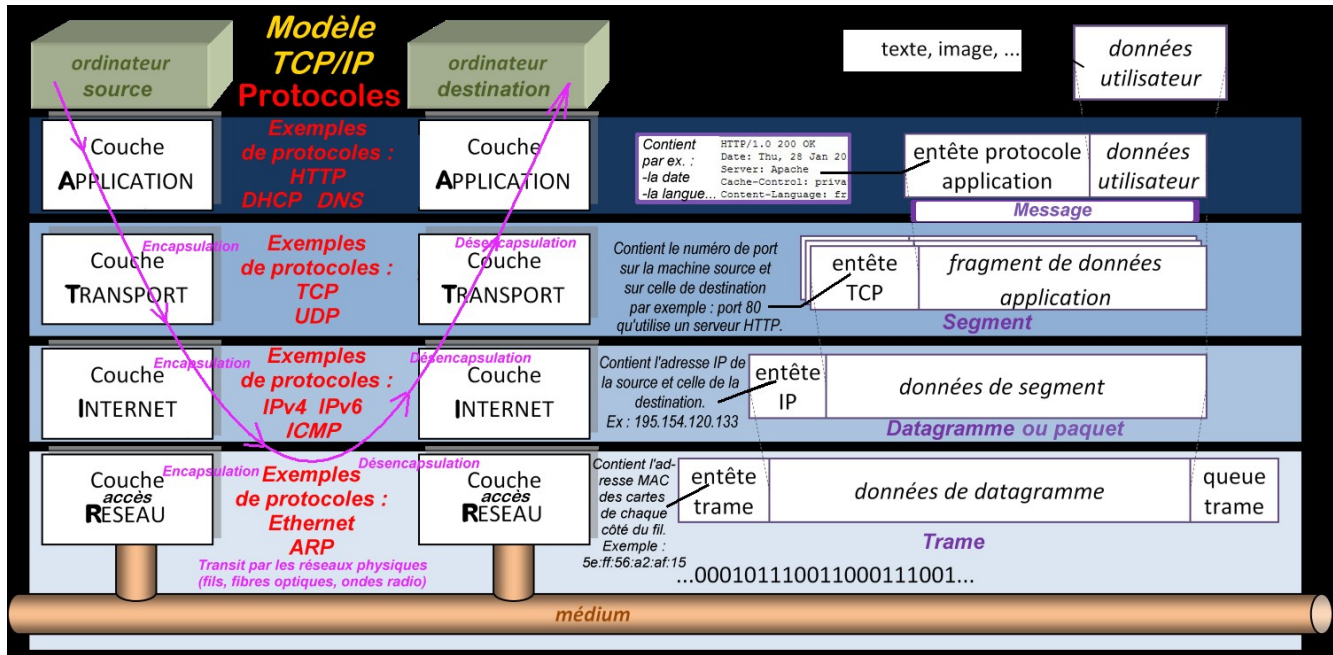
## Modèle TCP/IP

Les normes de communication des réseaux informatiques sont définies dans des " modèles ".

- Il existe un modèle général qui a 7 couches de protocoles. Il est nommé modèle **OSI** - *Open Systems Interconnexion* - que nous n'étudions pas ici.
- Le modèle de communication que nous étudions est le modèle **TCP/IP** - *Transmission Control Protocol / Internet Protocol* -. Il a 4 couches de protocoles.
  - La couche du haut est la **couche application**. Les protocoles qu'on y trouve sont des ensembles de règles qui concernent par exemple le transfert de fichiers, de courrier électronique, la gestion des liens hypertexte, la connexion sécurisée, la recherche de l'adresse IP correspondant à une URL.
  - A l'avant dernier étage, on trouve la **couche transport**. Son rôle est de maintenir de la conversation entre deux machines.
  - En dessous, on trouve la **couche internet**. Son rôle est d'acheminer les paquets d'information entre deux machines à travers un réseau de réseaux.
  - En bas, on trouve la **couche réseau** qui est proche du matériel. Elle permet d'envoyer des paquets IP en utilisant des cartes réseau et des câbles ou des ondes radio.

**Schéma du modèle TCP/IP avec des exemples de protocoles pour chacune des quatre couches.**

**Remarquez que les noms donnés aux morceaux d'informations changent selon la couche : Message, Segment qui contient le message, Paquet qui contient le segment, Trame qui contient le paquet.**



### Explication des acronymes :

#### Couche Application :

**HTTP :** HyperText Transfer Protocol est un protocole de communication client-serveur développé pour le Web.

**DHCP :** Dynamic Host Configuration Protocol est un protocole qui configure automatiquement les adresses IP.

**DNS :** Domain Name System : traduit l'adresse (par exemple *www.google.fr*) en adresse IP

#### Couche Transport :

**TCP :** Transmission Control Protocol est un protocole qui découpe le flux d'octets en segments.

**UDP :** User Datagram Protocol : ressemble à TCP, mais sans s'assurer de la bonne réception des segments.

#### Couche Internet :

**IP :** Internet Protocol : encapsule les segments TCP en ajoutant les adresses IP. Actuellement IPv4 évolue en IPv6.

**ICMP :** Internet Control Message Protocol : permet le contrôle des erreurs de transmission (par ex : faire un ping)

#### Couche Réseau :

**Ethernet :** Protocole de réseau local : encapsule les paquets IP en ajoutant les adresses MAC des cartes réseau.

**ARP :** Address Resolution Protocol : traduit l'adresse IP en adresse MAC d'un poste situé sur le même sous-réseau.

([http://www.astrovirtuel.fr/jupyter/19\\_pnsi\\_cours/07.reseaux\\_couches.png](http://www.astrovirtuel.fr/jupyter/19_pnsi_cours/07.reseaux_couches.png))

- Visualisez la vidéo suivante puis répondez aux questions 12) et 13) :



[http://www.astrovirtuel.fr/jupyter/19\\_pnsi\\_cours/tcp\\_ip.mp4](http://www.astrovirtuel.fr/jupyter/19_pnsi_cours/tcp_ip.mp4)

12) On suppose que du côté source (c'est à dire côté émetteur), on veut demander une page Web au serveur (c'est à dire sur l'ordinateur de destination).

- Le navigateur prépare donc une requête selon le protocole HTTP (autrement dit un message qui demande la page Web).
- Ce message de requête est découpé en des segments TCP numérotés selon le protocole **TCP**.
- Les segments sont à leur tour mis dans des paquets selon le protocole **IP**. Chaque paquet IP porte l'adresse de la machine d'émission et de la machine de destination.

Quel est le nom de la couche dans laquelle on trouve le protocole IP ?

Réponse :

\*13) Dans la couche au-dessus - la couche de Transport -, on utilise le protocole TCP si on veut s'assurer de la bonne réception de tous les segments. Quelle autre protocole peut-on utiliser lorsqu'on n'a pas besoin de vérifier la bonne réception de tous les segments ?\*

Réponse :

- A présent, regardez une courte vidéo où le fonctionnement des protocoles TCP et IP est expliqué par une ingénieure de chez *Spotify* (entreprise spécialisée dans la distribution de musique sur Internet) et par Vinton Cerf co-inventeur du protocole TCP/IP :



[http://www.astrovirtuel.fr/jupyter/19\\_pnsi\\_cours/vint\\_cerf.mp4](http://www.astrovirtuel.fr/jupyter/19_pnsi_cours/vint_cerf.mp4)

## 2.2 Les protocoles TCP et IP

Lisez le paragraphe **Les protocoles TCP et IP** p. 192  
puis complétez :

\*14) A l'arrivée, chez le destinataire, quel protocole permet de reconstituer le message envoyé à partir des segments reçus ?\*

Réponse :

\*15) Quel protocole utilise-t-on à la place de **\*\*HTTP\*\*** pour sécuriser un transfert sur le Web ?\*

Réponse :

## 3. Rôle des différents protocoles de communication

### 3.1 Le service Système de Noms de Domaine

Lisez le paragraphe **Rôle des différents protocoles de communication** p. 193 et la première moitié de la p. 194

puis complétez :

\*16) On nomme **URL** une adresse web telle que education.gouv.fr  
Que signifie l'acronyme URL ?\*

Réponse :

\*17) Les serveurs **DNS** (Domain Name System) sont des serveurs spéciaux qui, à une l'URL d'une ressource, font correspondre l'adresse IP d'une machine.  
Ouvrez un nouvel onglet dans votre navigateur et tapez l'adresse IP 216.58.198.195  
Vous vous retrouvez directement sur la page d'accueil d'un site. Lequel ?\*

Réponse :

\*18) Quel protocole de la couche Application permet, à partir d'une adresse web, d'obtenir l'adresse IP (et donc permet par exemple de traduire google.com en 216.58.198.195) ?\*

Réponse :

- Voici un tableau qui résume les informations ajoutées par chaque couche de protocole lors de l'envoi d'une requête (demande) avec le protocole DNS. Par exemple l'utilisateur saisit dans la barre d'adresse de son navigateur la chaîne de caractères **google.com** et appuie sur la touche Entrée. On retrouve les quatre couches du modèle TCP/IP : Application, Transport, Internet, Réseau :

Couche	Protocole	Nombre d'octets ajoutés ici	Cumul des octets ajoutés
Application	DNS	32 octets qui contiennent le nom de domaine et le type d'adresse.	32
Transport	UDP	8 octets qui contiennent les numéros de port de la source et de destination.	40
Internet	IP	20 octets qui contiennent les numéros IP de la source et de destination.	60
Réseau	Ethernet	14 octets qui contiennent les adresses MAC des cartes réseau sur le "fil" qui les relie.	74

\*19) La réponse, en provenance d'un serveur DNS sur Internet, revient avec les mêmes informations et, en plus, l'adresse IP demandée, par exemple **216.58.198.195**. Combien cela fait-il d'octets en plus ?\*

Réponse :



- Une fois l'adresse IP du destinataire connu, l'utilisateur peut utiliser le site, en cliquant par exemple sur des liens hypertexte. Cette fois le protocole HTTP est utilisé au niveau Application et le protocole TCP au niveau Transport.

Couche	Protocole	
Application	HTTP	.
Transport	TCP	.
Internet	IP	
Réseau	Ethernet	La MTU (Maximum Transmission Unit) = 1500 octets

\*20) Quel protocole joue le même rôle qu'Ethernet au niveau couche Réseau, mais en permettant l'utilisation d'ondes radio à la place des fils ?\*

Réponse :

\*21) Quels protocoles, qui sont au-dessus de la couche Transport, permettent de sécuriser les échanges de données confidentielles ?\*

Réponse :

\*22) Qu'est-ce qu'un serveur renvoie au navigateur à la demande d'accès à un site sécurisé ?\*

Réponse :

\*23) Lorsqu'un serveur reçoit une requête d'un ordinateur "client", quelles informations a-t-il sur le client ? (citez 3 informations)\*

Réponse :

## 3.2 Récupération de paquets

Lisez le paragraphe **Récupération de paquets** p. 194 et première moitié de la p. 195 puis répondez aux questions suivantes :

\*24) Un émetteur A envoie un message à un récepteur B.\*

\* B, à son tour, va émettre le message : "A, j'ai bien reçu ton message".\*

\*Comment nomme-t-on ce message ?\*

Réponse :

\*25) Un protocole particulier est utilisé dans la plus basse couche (la couche réseau) pour garantir la bonne réception par B de toutes les trames envoyées par A.\*

\*Voici un exemple de fonctionnement avec perte de certaines trames à cause de problèmes sur le câble ou sur la liaison radio :\*

- A : "J'envoie la trame n avec le bit "drapeau" ("flag" en anglais) 0 ."
- **Par exemple cette trame est perdue.** Donc B ne peut pas envoyer de trame d'acquittement.
- A n'a pas reçu d'acquittement. Au bout d'un certain délai :
- A : "J'envoie la trame n avec le bit drapeau 0 ."
- Cette trame est reçue par B.
- B : "Trame d'acquittement : bien reçu une trame avec le drapeau 0 ."
- Cette trame d'acquittement est reçue par A.
- A en déduit qu'il peut envoyer la trame suivante avec un drapeau 1 .
- A : "J'envoie la trame n + 1 avec le drapeau 1 ."
- Cette trame est reçue par B.
- B : "Trame d'acquittement : bien reçu une trame avec le drapeau 1 ."
- **Par exemple cette trame est perdue.**
- A n'a pas reçu d'acquittement. Au bout d'un certain délai :
- A : "J'envoie la trame n + 1 avec le drapeau 1 ."
- Cette trame est reçue par B.
- B : "Trame d'acquittement : bien reçu une trame avec le drapeau 1 ."
- Cette trame d'acquittement est reçue par A.
- A en déduit qu'il peut envoyer la trame suivante avec un drapeau 0 .
- A : "J'envoie la trame n + 2 avec le drapeau 0 ."
- A continue d'envoyer la trame n + 2 tant qu'il n'a pas reçu la trame d'acquittement avec le drapeau 0 .

\*Comment appelle-t-on ce protocole ?\*

Réponse :

### 3.3 Adresse IP

Lisez le paragraphe **Adresse IP** p. 195, 196 et 197  
puis répondez aux questions suivantes :

\*26) Dans la cellule ci-dessous, entrez le code suivant :\*

```
import socket # le module socket permet de créer des connexions reseau en-dessous de TCP
(inclut donc IP).

print(socket.gethostbyname("python.org"))
```

In [ ]:

\*Qu'a-t-on obtenu par la fonction `gethostbyname` en donnant `python.org` comme nom d'hôte ?\*

Réponse :

## Remarque sur sockets

Le mot `socket` a deux sens différents :

- Un socket matériel est un connecteur utilisé pour brancher un microprocesseur sur une carte mère.
- Dans le cadre de ce chapitre, un socket est une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et d'une manière uniforme les services d'un protocole réseau. Il lui sera par exemple aisé d'établir une session TCP entre un poste "client" et un poste "serveur", puis de recevoir et d'expédier des données grâce à elle.

\*27) Dans la cellule ci-dessous, entrez le code suivant :\*

```
import socket

print(socket.getaddrinfo("python.org", 80, proto=socket.IPPROTO_TCP))
```

In [ ]:

\*Le fonction `getaddrinfo` de la bibliothèque `socket` va chercher les informations de connexion TCP sur la machine associée au nom de domaine `python.org` concernant le port 80. Combien d'éléments contient le tuple d'informations qui est retourné ?\*

Réponse :

## Remarque sur les ports logiciels

- Un n° de port logiciel peut être vu comme le n° d'une "trappe" ouverte par laquelle la machine distante va communiquer par les protocoles de la couche transport (TCP ou UDP).
- Le n° de port dépend du type d'application.
- Certains numéros de port sont réservés. Par exemple pour HTTP c'est le port 80, pour DNS c'est le port 53.
- Pour d'autres utilisations, par exemple pour faire communiquer deux programmes, le programmeur peut choisir lui-même le numéro de port parmi les numéros en dehors de [ceux qui ont été assignés](https://fr.wikipedia.org/wiki/Liste_de_ports_logiciels) ([https://fr.wikipedia.org/wiki/Liste\\_de\\_ports\\_logiciels](https://fr.wikipedia.org/wiki/Liste_de_ports_logiciels)) pour des utilisations spécifiques, notamment par l'Internet Assigned Numbers Authority (IANA).

\*28) Dans la cellule ci-dessous, entrez le code suivant :\*

```
import socket

print(socket.getaddrinfo("example.org", 80, proto=socket.IPPROTO_TCP))
```

In [ ]:

\*Cette fois, les informations sont retournées sous la forme d'un tuple de deux tuples à 5 éléments :\*

\*Quelle type d'adresse IP apparait dans le premier des deux tuples ?\*

Réponse :

\*29) Le méthode `gethostbyaddress` de la bibliothèque `socket` va chercher le nom d'hôte primaire sur la machine qui a l'adresse IP passée en argument.\*

\*Dans la cellule ci-dessous, entrez le code suivant :\*

```
import socket

print(socket.gethostbyaddr("8.8.8.8"))
```

In [ ]:

\*Quelle compagnie met à disposition à l'adresse 8.8.8.8 son serveur DNS ?\*

Réponse :

## Pour plus d'informations sur les fonctions Python :

On peut consulter par exemple le site <https://docs.python.org/fr/3.5/genindex.html> (<https://docs.python.org/fr/3.5/genindex.html>). Par exemple, pour avoir plus d'informations sur la fonction `gethostbyaddress` du module `socket`, allez à la lettre G et cherchez `gethostbyaddress`.

## Une adresse IPv4 se décompose en deux parties

Dans l'adresse IPv4 d'une machine, par exemple 192.168.1.13, si on sait que le masque de réseau est 255.255.255.0, alors on obtient l'identifiant du réseau par un **and** bit à bit entre l'adresse de la machine et le masque.

Vocabulaire : Ce réseau particulier sera probablement relié à Internet et formera donc un sous-réseau d'Internet. On dira donc plutôt que 255.255.255.0 est un **masque de sous-réseau**.

En effet, en écrivant en binaire l'adresse de la machine et le masque, et en faisant un *and* bit à bit entre l'adresse et le masque on a :

Adresse machine	192								168								1								13							
	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	
Masque de réseau	255								255								255								0							
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0		
Identifiant du réseau = Adresse machine and Masque	1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0			
	192								168								1								0							

Un tel masque laisse au réseau les 3 premiers octets (puisque le masque commence par 255.255.255) et il ne laisse qu'un seul octet pour numéroter les différentes machines sur ce réseau local.

Cela ne fait donc que 256 machines possibles de 192.168.1.0 à 192.168.1.255.

Mais les deux valeurs extrêmes, ici 192.168.1.0 et 192.168.1.255, sont réservées.

La première 192.168.1.0 est l'identifiant du réseau lui-même.

La dernière 192.168.1.255 est l'adresse à utiliser pour la diffusion à toutes les machines du réseau. On l'appelle adresse de **broadcast**.

Cela laisse la possibilité d'adresser 254 machines (on dit des "hôtes") sur ce réseau local de 192.168.1.1 à 192.168.1.254.

Dans notre exemple, la machine est identifiée par 0.0.0.13 sur ce réseau local.

**En résumé :**

**Adresse machine : 192.168.1.13**

**Masque de sous-réseau : 255.255.255.0**

On en déduit que :

**Identifiant du réseau (NET ID) : 192.168.1.0**

**Identifiant de l'hôte, c'est à dire d'une machine connectée au réseau (HOST ID) : 0.0.0.13**

Le réseau donné dans l'exemple précédent a :

- Une adresse IP qui commence par 192
- Un masque de sous-réseau égal à 255.255.255.0

C'est un cas particulier de réseau de classe C.

De façon générale, un réseau de classe C est caractérisé par :

- Une adresse IP qui commence par un nombre pouvant aller de 192 et 223
- Un masque de sous-réseau égal à 255.255.255.0

\*30) Complétez le tableau suivant :\*

Classe du réseau	Le 1er nombre de l'adresse du réseau est entre ... et ...	Le masque de réseau est ...	Le réseau peut contenir ... adresses
A	...	...	...
B	...	...	...
C	192 et 223	255.255.255.0	256

- On vient de voir qu'un réseau de classe C a un masque 255.255.255.0 ou encore, en binaire, 11111111 11111111 11111111 0000000. Ses 24 premiers bits sont à 1.

L'adresse IP de la machine identifiée par 0.0.0.13 sur ce réseau local peut être notée de façon commode :

**192.168.1.13/24** ce qui indique la taille du masque (ici les 24 premiers bits sont à 1).

Mais on peut choisir un masque de manière plus fine que ceux des classes A, B, C.

Par exemple, si une machine sur un certain réseau a pour adresse IP

**192.168.1.18/27** cela signifie que son masque est en binaire 11111111 11111111 11111111 11100000 (les 27 bits de gauche sont à 1).

**\*31) Donner la valeur du masque en notation décimale.\***

Indication : Dans le shell (c'est à dire l'interface qui permet d'écrire des lignes de code) de votre calculatrice TI83-CE Python ou dans la fenêtre ci-dessous, saisissez le code :

```
int("11111111",2) # La fonction int a deux arguments : Une chaine de caractères "chiffres" e
t
# la base dans laquelle est écrite le nombre.
```

puis

```
int("11100000",2)
```

In [ ]:

Réponse :

**Savoir-Faire :**

**\*Un hôte sur un réseau a comme adresse IP \*\*192.168.1.18/27\*\*.\***

- 1) Quel est l'identifiant de ce réseau ?
- 2) Quel est l'identifiant de l'hôte ?
- 3) Quelle est la plage d'adresses IP disponible pour les machines sur ce réseau ?

\*Méthode :\*

1) On commence par écrire **en binaire** le masque et l'adresse IP de l'hôte.

- le masque : /27 indique que les 27 bits de gauche sont à 1. Donc l'écriture sur 32 bits du masque est

```
11111111 11111111 11111111 11100000
```

- l'adresse IP de l'hôte est 192.168.1.18. La fonction Python **bin** permet de convertir un nombre en base 10 en base 2

```
bin(192)
```

```
11000000
```

```
bin(168)
```

```
10101000
```

```
bin(1)
```

```
00000001
```

```
bin(18)
```

```
00010010
```

Donc l'écriture sur 32 bits de l'adresse IP de l'adresse de l'hôte est

```
11000000 10101000 00000001 00010010
```

L'identifiant du réseau s'obtient en faisant un *and* bit à bit entre l'adresse IP de l'hôte et le masque ce qui donne :

Adresse de l'hôte	192								168								1								18							
	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	
Masque de réseau	255								255								255								224							
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	
Identifiant du réseau = Adresse de l'hôte and Masque	1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0		
	192								168								1								0							

Donc l'identifiant de ce réseau est 192.168.1.0

2) L'identifiant de l'hôte s'obtient en faisant un *and* bit à bit entre l'adresse IP de l'hôte et le

*masque* = 00000000.00000000.00000000.00011111 ce qui donne :

Adresse de l'hôte	192								168								1								18							
	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	
<i>masque</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1		
Identifiant de l'hôte = Adresse de l'hôte <b>and</b> <i>masque</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1		
	0								0								0								18							

Donc l'identifiant de l'hôte est 00000000 00000000 00000000 00010010 (autrement dit les 5 bits de droite dans l'adresse IP de l'hôte en binaire).

Ce qui donne en décimal : 0.0.0.18

3)

Les 27 bits de gauche étant réservés pour l'identifiant réseau (par le masque) on ne pourra disposer pour les machines que de la plage des 5 bits de droite de 00000 à 11111.

Autrement dit, la plage des adresses IP disponible pour les machines est :

11000000 10101000 00000001 00000000

jusqu'à

11000000 10101000 00000001 00011111

En convertissant en décimal, cela donne la plage depuis

192.168.1.0

jusqu'à

192.168.1.31

SAUF que la première adresse 192.168.1.0 est l'identifiant réseau et la dernière 192.168.1.31 est l'adresse de broadcast (à utiliser pour diffuser à toutes les machines connectées).

Donc la plage réellement disponible est de 192.168.1.1 à 192.168.1.30 soit 30 machines seulement.

### Adresses IPv6

Elles comportent 16 octets répartis en 8 groupes de 2 octets séparés par le signe : (deux-points).

Elles vont donc en théorie, en écriture hexadécimale, de

0000:0000:0000:0000:0000:0000:0000:0000

à

FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF



## Deux règles de notation

**Règle n°1** Les zéros à gauche de chaque groupe peuvent être omis.

Exemple :

2001:0660:7401:0200:0000:0000:0EDF:BDD7

peut s'écrire de façon abrégée :

2001:660:7401:200:0000:0000:EDF:BDD7

On serait tenté de noter encore de façon plus abrégée :

2001:660:7401:200:::EDF:BDD7

Mais il y a une autre règle :

## Règle n°2

La séquence "::" ne peut apparaître qu'une fois dans une adresse.

Donc dans notre exemple, on notera

2001:660:7401:200::EDF:BDD7

\*32) Comment note-t-on de façon abrégée l'adresse IPv6 0000:0000:0000:0000:0000:0000:0000:0001 ?\*

Réponse :

\*33) Une adresse IP avec le protocole IPv4 comporte combien d'octets ?\*

Réponse :

\*34) Quel est le nombre maximal d'adresses Internet publiques avec la version IPv4 ?\*

Réponse :

\*35) Ce nombre est-il suffisant ?\*

Réponse :

\*36) Pour augmenter le nombre maximal d'adresses publiques, IPv6 a pris la suite d'IPv4. Une adresse IP avec le protocole IPv6 comporte combien d'octets ?\*

Réponse :

\*37) Quel est le nombre maximal d'adresses Internet publiques avec la version IPv6 ?\*

Réponse :

## 4. Architecture d'un réseau

### 4.1 Différentes méthodes de connexion

Lisez le paragraphe **Architecture d'un réseau** p. 197 et 198  
puis répondez aux questions suivantes :

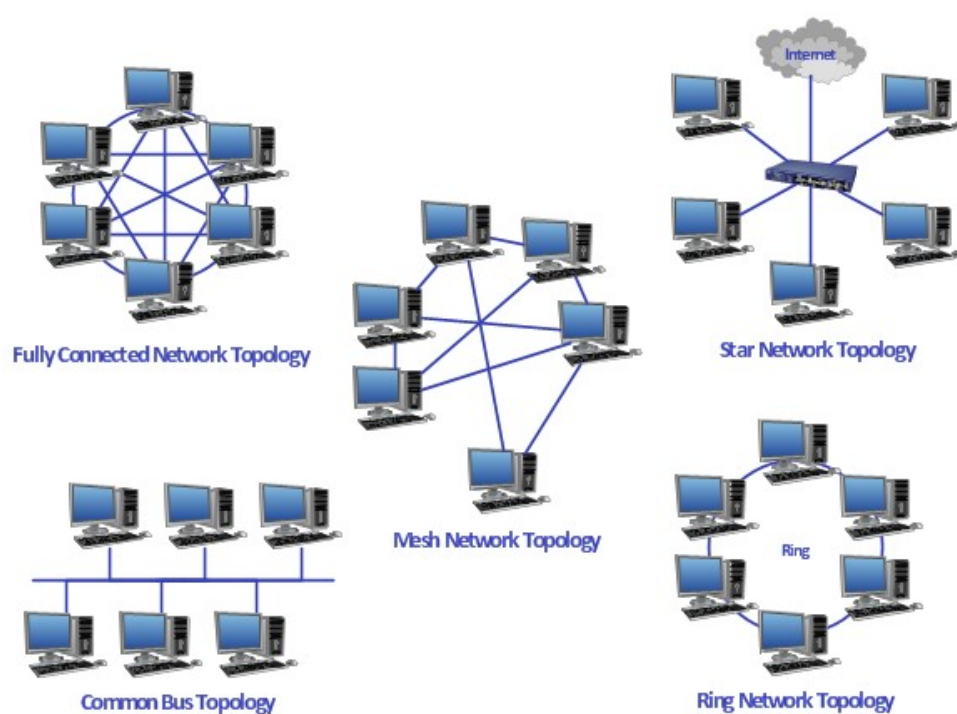
\*38) Que signifie " connexion peer-to-peer " ?\*

Réponse :

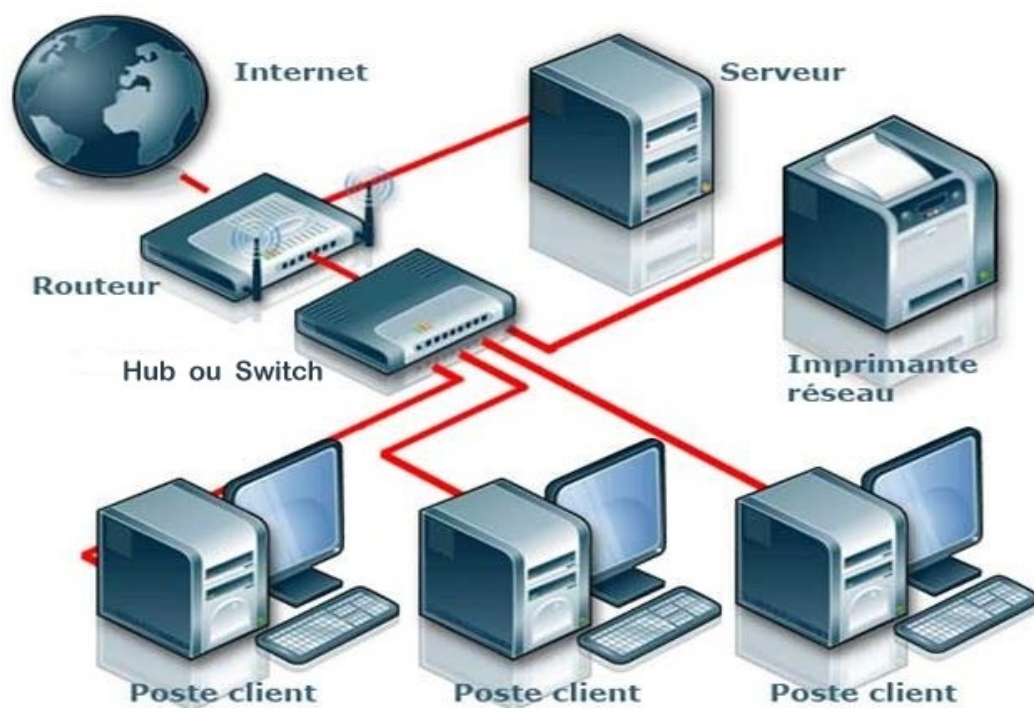
\*39) Que signifie " appareil client " ?\*

Réponse :

- Voici différentes topologies de réseaux : Fully connected (tous les postes sont reliés deux à deux) ; Mesh (réseau maillé) ; Star (en étoile) ; Common Bus (en bus) ; Ring (en anneau).



- Voici un réseau en étoile :



\*40) On suppose qu'un des postes clients envoie sur le réseau une requête vers l'adresse IP du Routeur.\*  
 \*Quelle différence cela fait-il d'avoir un Hub plutôt qu'un Switch ?\*

Réponse :

\*41) Quelle sorte d'adresse est associée à chaque port du switch afin que le switch ait connaissance de l'identité des machines qui lui sont reliées ?\*

Réponse :

- Sur un réseau local, les adresses IP peuvent être fixées (une adresse définie et permanente pour chaque poste). Mais il est préférable d'attribuer dynamiquement les adresses IP (c'est à dire quand un poste se connecte au réseau) . Cela évite les erreurs de saisie manuelle pour chaque poste. Cela permet aussi à des postes nomades (ordinateurs ou téléphones portables) de se connecter au réseau.

\*42) Par quel type de serveur les adresses dynamiques sont-elles attribuées ?\*

Réponse :

- Exemple de réseau chez un particulier :

Box avec Passerelle DHCPv4 192.168.1.1 et masque de sous-réseau 255.255.255.0 *Cela signifie que les postes connectés auront une adresse IP attribuée dynamiquement sur le réseau local du particulier situées entre 192.168.1.2 et 192.168.1.254 (sauf si certaines adresses statiques ont été paramétrées dans la box).*

État DHCPv4	Actif
Adresse IPv4 de début	192.168.1.2
Adresse IPv4 de fin	192.168.1.254
Masque de sous réseau IPv4	255.255.255.0
Bail DHCPv4	86400
Passerelle DHCPv4	192.168.1.1
Adresse MAC	08:D5:9D:BD:09:CC

Et on aura par exemple les adresses IP des appareils suivants :

- Ordinateur portable 192.168.1.12 (connecté par Wi-Fi)
- Smartphone 192.168.1.19 (connecté par Wi-Fi)
- Imprimante 192.168.1.14 (connecté par câble Ethernet)
- Décodeur télévision 192.168.1.13 (connecté par câble Ethernet)

qui sont attribuées dynamiquement par le serveur DHCP (logiciel inclus dans la box) et qui peuvent éventuellement changer d'une fois à une autre.

Remarque : il est possible de faire un réglage manuellement sur certains appareils (par exemple une imprimante) et sur la box si on veut que telle ou telle adresse IP soit **statique** autrement dit qu'elle ne change jamais.

## 4.2 Quelques lignes de commande Linux

Lisez le paragraphe **Ligne de commande** du bas de la p. 198 à la p. 200 puis répondez aux questions suivantes :

### 4.2.1 ifconfig

- Ouvrir le terminal Linux *par un seul clic* puis saisissez, pour voir la configuration des interfaces réseau :

```
$ ifconfig
```

On voit apparaitre un texte ressemblant à :

```
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:d0:24:55 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.12 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::5308:ae80:e801:d4d6 prefixlen 64 scopeid 0x20<link>
    inet6 2a01:cb05:8315:cb00:db98:d1f6:d6a0:e98 prefixlen 64 scopeid 0x0<global>
    ether b8:27:eb:85:71:00 txqueuelen 1000 (Ethernet)
    RX packets 6510 bytes 7749809 (7.3 MiB)
    RX errors 0 dropped 45 overruns 0 frame 0
    TX packets 2513 bytes 310620 (303.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Il y a trois zones :

- **eth0** concerne la carte Ethernet où on peut brancher une prise RJ45. Dans notre cas, l'adresse MAC de cette carte est b8:27:eb:d0:24:55 (6 octets écrits en hexadécimal). L'adresse MAC est liée au matériel. L'adresse MAC de la carte Ethernet de ce Raspberry lui est propre et restera toujours la même.
- **lo** pour *loopback*. C'est une interface virtuelle qui permet à l'ordinateur de communiquer avec lui-même. Son intérêt est de pouvoir faire des tests en réseau lors de la mise au point de logiciels, même si aucun réseau réel n'est disponible. L'adresse de la machine locale est 127.0.0.1 et son nom est **localhost**.
- **wlan0** pour Wireless Local Area Network (réseau local sans fil). C'est l'interface Wi-Fi par ondes radio. Elle possède aussi une adresse MAC unique et liée au matériel. Dans notre cas, c'est b8:27:eb:85:71:00. On voit qu'en plus il y a une adresse IPv4 sur 4 octets 192.168.1.12 avec un masque de sous-réseau 255.255.255.0 et une adresse de diffusion à toutes les machines du réseau (broadcast) 192.168.1.255. Par ailleurs, on peut voir que cette liaison est compatible avec des adresses IPv6 sur 16 octets.

\*43) Copiez - collez dans la zone ci-dessous le résultat de votre commande\*

```
$ ifconfig
```

Réponse :

\*44) Relevez ci-dessous les informations que vous lisez pour votre Raspberry :\*

Réponse :

- Carte Ethernet (à lire à la rubrique eth0)
  - Adresse MAC (6 octets à lire à côté de ether) : ...
- Loopback
  - Adresse IP de l'hôte local : ...
  - Masque de réseau : ...
- Wi-Fi (à lire à la rubrique wlan0) :
  - Adresse IP attribuée à l'interface Wi-Fi : ...
  - Masque de réseau : ...
  - Adresse MAC (6 octets à lire à côté de ether) : ...

\*45) Relevez ci-dessous les informations du Raspberry de votre voisin :\*

Réponse :

- Wi-Fi :
  - Adresse IP attribuée à l'interface Wi-Fi : ...

## 4.2.2 ping

\*46) Quelle commande permet d'envoyer une demande d'écho à une machine distante dont on connaît l'adresse IP ?\*

Réponse :

\*47) La commande ping utilise un protocole particulier. Lequel ?\*

Réponse :

\*48) Dans le terminal, saisissez la commande pour demander un ping sur le site **education.gouv.fr** :\*

```
$ ping education.gouv.fr
```

Pour arrêter cette commande, appuyez sur la Ctrl et en maintenant la touche Ctrl (contrôle) appuyée, appuyez sur C. En bas, on peut lire le nombre de paquets transmis, le nombre de paquets reçus, le pourcentage de paquets perdus et le temps total de transmission de tous ces paquets.

Sur la dernière ligne intitulée rtt (Round-trip time qui est le temps d'aller-retour), vous pouvez voir les statistiques.

Dans l'ordre on lit le temps le temps minimal, le temps moyen, le temps maximal, l'écart-type.

\*Notez ci-dessous la valeur que vous avez obtenue pour le temps moyen d'aller-retour pour le site education.gouv.fr :\*

Réponse :

\*49) Quelle valeur de TTL (Time-to-live ou "durée de vie") avez-vous obtenue ?\*

Réponse :

\*50) En déduire le nombre de routeurs traversés\*

Réponse :

\*51) Dans le terminal, saisissez la commande pour demander un ping sur l'adresse IP du Raspberry de votre voisin:\*

```
$ ping NNN.NNN.NNN.NNN où NNN.NNN.NNN.NNN est son adresse IP.
```

Pour arrêter cette commande, appuyez sur la Ctrl et en maintenant la touche Ctrl (contrôle) appuyée, appuyez sur C. En bas, on peut lire le nombre de paquets transmis, le nombre de paquets reçus, le pourcentage de paquets perdus et le temps total de transmission de tous ces paquets.

Sur la dernière ligne intitulée rtt (Round-trip time qui est le temps d'aller-retour), vous pouvez voir les statistiques.

Dans l'ordre on lit le temps le temps minimal, le temps moyen, le temps maximal, l'écart-type.

\*Notez ci-dessous la valeur que vous avez obtenue pour le temps moyen d'aller-retour pour le raspberry de votre voisin :\*

Réponse :

\*52) Puisque vous connaissez l'adresse IP du Raspberry de votre voisin et que la liaison fonctionne (retour des 'ping') profitez-en pour aller chercher le nom d'hôte de son Raspberry :\*

\*Dans la cellule ci-dessous, entrez le code suivant :\*

```
import socket

print(socket.gethostbyaddr("NNN.NNN.NNN.NNN"))
```

\*Quel est son nom d'hôte ?\*

Réponse :

### 4.2.3 traceroute

\*53) Dans le terminal, saisissez la commande pour demander un traceroute sur le serveur \*\*education.gouv.fr\*\* :\*

```
$ traceroute education.gouv.fr
```

On obtient une liste de routeurs par où passent les paquets IP pour arriver au serveur **education.gouv.fr**.

Après un certain nombre de routeurs, on obtient uniquement \*\*\*. On peut conclure qu'à ce niveau, nous n'avons plus de réponse des routeurs aux paquets ICMP envoyés dans un temps raisonnable. Un maximum de 30 sauts (routeurs) peut être affiché.

\*Combien de routeurs (c'est à dire de lignes écrites) pouvez-vous voir avec un traceroute education.gouv.fr ?\*



Réponse :

\*54) Dans le terminal, saisissez la commande pour demander un traceroute sur le serveur australien **www.atsb.gov.au**.

\*Combien de routeurs cette fois pouvez-vous voir ?\*

Réponse :

\*55) Pourquoi les paquets IP ont-ils plus de routeurs à traverser pour atteindre le serveur **www.atsb.gov.au** que le serveur **education.gouv.fr** ?\*

Réponse :

## 4.2.4 getent hosts

Sous Linux, la commande :

```
$ getent hosts
```

permet d'afficher le nom de domaine en fonction de l'adresse IP associée et réciproquement.

\*56) Dans le terminal, saisissez la commande pour demander un getent hosts sur le nom du serveur **education.gouv.fr** :

```
$ getent hosts education.gouv.fr
```

\*Quel est l'adresse IP correspondant au nom de domaine education.gouv.fr ?\*

Réponse :

\*57) Dans le terminal, saisissez la commande pour demander un getent hosts sur **l'adresse IP** que vous avez obtenue à la question 56)\*

\*Quel est le nom de l'hôte ?\*

Réponse :