

1 Notion de sous-réseau

Un réseau peut être composé de plusieurs sous-réseaux à base d'équipements matériels. Les machines d'un même sous-réseau possèdent le même identifiant, calculable à partir de l'adresse IP et du masque de sous-réseau.

Un sous-réseau est une subdivision logique d'un réseau de taille plus importante. Le masque de sous-réseau permet de distinguer la partie de l'adresse commune à tous les appareils du sous-réseau et celle qui varie d'un appareil à l'autre.

On peut aussi distinguer les réseaux en terme « géographique » :

- LAN, Local Area Network : réseau local d'assez petite dimension (ex : maison, lycée...).
On parle de WLAN si les machines sont connectées en Wi-Fi.
- WAN, Wide Area Network : réseau de grande envergure (ex : pays, internet...).

Un sous-réseau correspond typiquement à un réseau local sous-jacent.

Historiquement, on appelle également sous-réseau chacun des réseaux connectés à Internet.

2 Masque de sous-réseau

2.1 Principe

Les adresses IPv4 sont composées de deux parties : le sous-réseau et l'hôte.

Une écriture complète prend la forme suivante : 192.168.1.2/24 (notation CIDR, Classless Inter-Domain Routing).

Le masque de sous-réseau est un masque distinguant les bits d'une adresse IPv4 utilisés pour identifier le sous-réseau de ceux utilisés pour identifier l'hôte.

Dans l'exemple précédent, les 24 premiers bits désignent le réseau et les 8 derniers identifient l'hôte.

Le masque de sous-réseau peut utiliser la même représentation que celles des adresses IPv4 (codée sur 4 octets, soit 32 bits, représentés en notation décimale à point).

On utilise en pratique des masques constitués sous leur forme binaire d'une suite de 1 suivis d'une suite de 0 (ex : 11111111.11100000.00000000.00000000, soit 255.224.0.0 en décimal).

L'adresse du sous-réseau est obtenue en appliquant l'opérateur binaire ET bit à bit entre l'adresse IPv4 et le masque de sous-réseau. L'adresse de l'hôte à l'intérieur du sous-réseau est quant à elle obtenue en appliquant l'opérateur ET entre l'adresse IPv4 et le *complément à un* du masque (masque en inversant les 1 et les 0).

Deux adresses IP appartiennent à un même sous-réseau si elles ont en commun les bits du masque de sous-réseau.

2.2 Exemple

Soit une machine d'adresse IPv4 192.168.1.2, appartenant au réseau de masque 255.255.255.0 ; soit en binaire :

- adresse 11000000.10101000.00000001.00000010
- masque 11111111.11111111.11111111.00000000

L'adresse du réseau est donc :

$$\begin{array}{rcl}
 & 11000000.10101000.00000001.00000010 & (192.168.1.2) \\
 \& & \\
 \& 11111111.11111111.11111111.00000000 & (255.255.255.0) \\
 = & 11000000.10101000.00000001.00000000 & (192.168.1.0)
 \end{array}$$

Celle de la machine (hôte) est :

$$\begin{array}{rcl}
 & 11000000.10101000.00000001.00000010 & (192.168.1.2) \\
 \& & \\
 \& 00000000.00000000.00000000.11111111 & (0.0.0.255) \\
 = & 00000000.00000000.00000000.00000010 & (0.0.0.2)
 \end{array}$$

3 Exercices

3.1 Extraction à la main

Vérifier (à la main) qu'une machine d'adresse IPv4 91.198.174.2/19 appartient au sous-réseau 91.198.160.0 et son identifiant d'hôte est 0.0.14.2.

3.2 Extraction en Python

Écrire une fonction en Python qui extrait la partie sous-réseau et la partie hôte d'une adresse IP passée en paramètre sous forme de chaîne de caractères au format "a.b.c.d/x" (a, b, c, et d sont des nombres entre 0 et 255, et x est un nombre entre 1 et 30).

Pistes de travail :

Beaucoup de manières d'arriver au résultat sont possibles !

On conseille de découper le problème en sous-fonctions intermédiaires.

On pourra par exemple s'intéresser aux méthodes de chaînes de caractères, aux opérateurs bitwise en Python (opérateur `&` ou `>>` par exemple), aux conversions binaire/décimale...

Ces pistes ne sont pas obligatoires, mais proposent des idées que l'on peut creuser.

4 Correction

4.1 Extraction à la main

L'adresse IPv4 est 91.198.174.2/19.

On peut d'abord convertir le masque de sous-réseau sous forme binaire puis décimale pointée :

- binaire : 11111111.11111111.11100000.00000000
- notation décimale pointée : 255.255.224.0

On peut aussi obtenir le masque correspondant à l'adresse de l'hôte (complément à 1 du masque de sous-réseau) :

- binaire : 00000000.00000000.00011111.11111111
- notation décimale pointée : 0.0.31.255

L'adresse du réseau est donc :

```

01011011.11000110.10101110.00000010   (91.198.174.2)
& 11111111.11111111.11100000.00000000   (255.255.224.0)
= 01011011.11000110.10100000.00000000   (91.198.160.0)

```

Celle de la machine (hôte) est :

```

01011011.11000110.10101110.00000010   (91.198.174.2)
& 00000000.00000000.00011111.11111111   (0.0.31.255)
= 00000000.00000000.00001110.00000010   (0.0.14.2)

```

4.2 Extraction en Python

Fonctions intermédiaires de conversion de l'écriture d'une adresse au format décimal pointé, au format d'entier.

```

def pointe2n(adr_pointe):
    """ Convertit une adresse décimale pointée en entier """
    a, b, c, d = [int(i) for i in adr_pointe.split('.')]
    a, b, c = a<<24, b<<16, c<<8
    return a + b + c + d

```

Remarque préliminaire :

```

0b11111111000000000000000000000000 = 2139095040
0b00000000111111110000000000000000 = 16711680
0b00000000000000001111111100000000 = 65280
0b00000000000000000000000011111111 = 255

```

```

def n2pointe(n):
    """ Convertit un entier en une adresse décimale pointée """
    A = (n & 2139095040) >> 24
    B = (n & 16711680) >> 16
    C = (n & 65280) >> 8
    D = n & 255

```

```
adr_pointe = '.'.join([str(A), str(B), str(C), str(D)])  
return adr_pointe
```

Autre fonction intermédiaire pour écrire un masque de sous-réseau (ou d'hôte) sous forme d'entier à partir de la longueur du nombre de bits de valeur 1 dans le masque de sous-réseau.

```
def masque(x):  
    """ Renvoie un masque de sous-réseau et d'hôte sous forme d'entier  
    en fonction du nombre de bits x de valeur 1 du masque de sous-réseau """  
    masque_reseau = int('0b' + '1' * x + '0' * (32-x), 2)  
    masque_hote = int('0b' + '1' * (32-x), 2)  
    return masque_reseau, masque_hote
```

Fonction principale :

```
def ip(adresse):  
    """ Extrait l'identifiant de réseau et d'hôte d'une adresse IPv4  
    au format "a.b.c.d/x".  
    Entrée : string "a.b.c.d/x".  
    Sortie : tuple de string  
    ("A.B.C.D" définissant le réseau, "K.L.M.N" définissant l'hôte).  
    """  
    # extrait la partie adresse et la partie taille du masque  
    adr_pointe, x = adresse.split('/')  
    x = int(x)  
    # conversion de l'adresse pointée en entier  
    n = pointe2n(adr_pointe)  
    # obtention des masques sous forme d'entier  
    masque_reseau, masque_hote = masque(x)  
    # extraction des adresses par application des masques  
    adr_hote = n & masque_hote  
    adr_reseau = n & masque_reseau  
    return n2pointe(adr_reseau), n2pointe(adr_hote)  
  
>>> adresse = "91.198.174.2/19"  
>>> ip(adresse)  
( '91.198.160.0', '0.0.14.2' )
```