

Cours 7 : Adressage IPv4

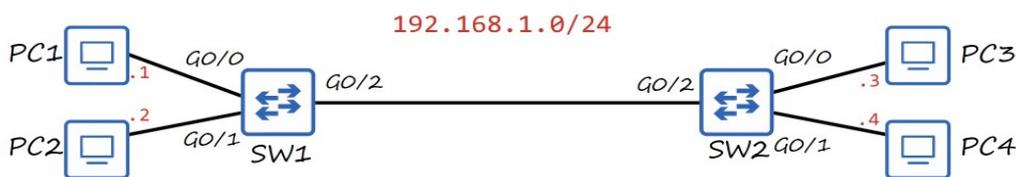
Dans les cours précédent nous avons vu comment fonctionnait un réseau local avec les PC connectés aux switches. Dans ce cours nous allons voir comment le trafic réseau est orienté en dehors du LAN à l'extérieur du réseau local vers d'autres LAN.

Pour cela la couche 3 est utilisé.

Revoyons donc rapidement quelques caractéristiques de la couche 3 (couche Réseau) :

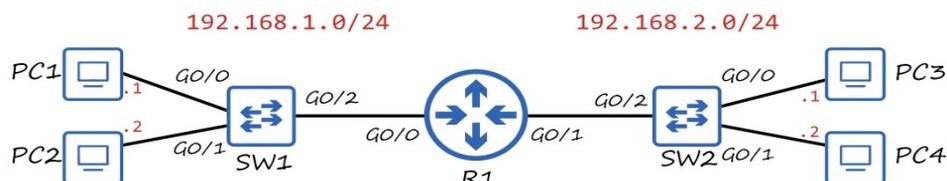
- Fournit une connectivité entre les hôtes de différents réseaux (en dehors du LAN)
- Fournit un adressage logique (IP adresses)
- Fournit une sélection de chemins entre la source et la destination, car sur internet il existe plusieurs chemins possibles pour atteindre une destination
- Les routeurs fonctionnent à la couche 3

Prenons l'exemple du schéma suivant :



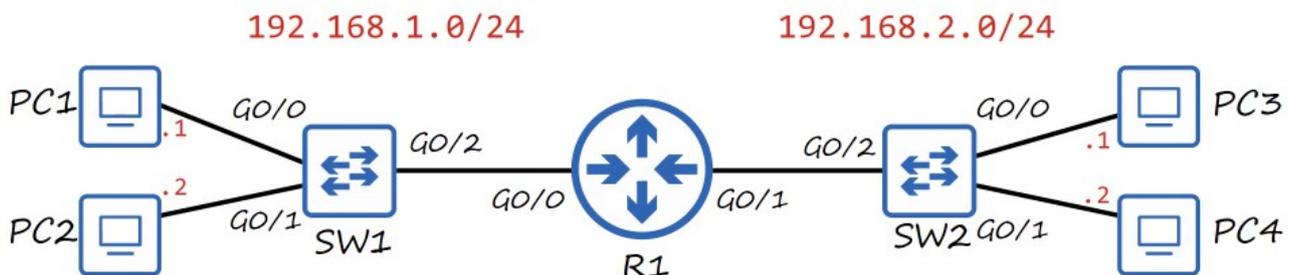
Nous avons vu auparavant que lorsqu'un PC veut connaître les adresses d'un réseau local il va envoyer des requêtes en Broadcast sur tout le réseau.

Mais que se passe-t-il si l'on ajoute un routeur entre les deux switches ?



A présent au lieu d'y avoir 1 seul réseau le réseau est divisé en 2 avec deux adresses sur deux réseaux différents : 192.168.1.0/24 ; 192.168.2.0/24

Le /24 est pour indiquer quelle est l'adresse fixe du réseau qui ne peut pas être changé avec : 192.168.1 ; le 0 est le chiffre qui change



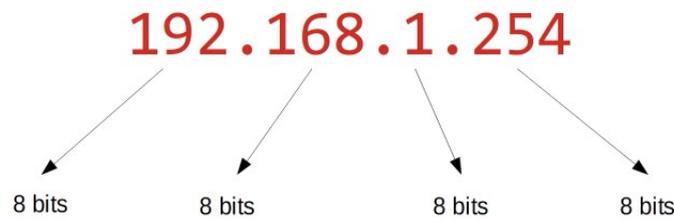
Une dernière chose à ajouter sur le schéma est l'adresse IP des interfaces du routeur. Pour G0/0 se sera : 192.168.1.254 et pour G0/1 se sera : 192.168.2.254

Cette fois si le PC1 envoie une requête sur tout le réseau la trame sera distribuée sur toutes les interfaces du switch jusqu'à l'interface du routeur sur son même réseau, mais la requête n'ira pas sur l'interface local de l'autre routeur.

Ceci est une capture d'écran d'une trame réseau prise sur Wikipédia :

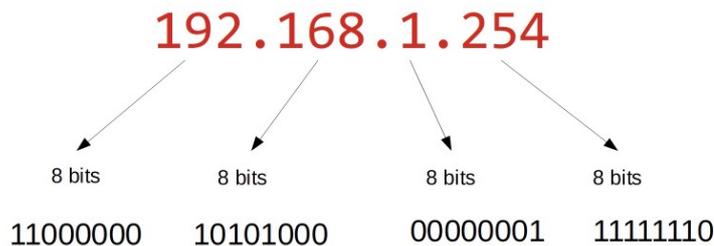
		IPv4 Header Format																															
Offsets	Octet	0				1				2				3																			
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP				ECN				Total Length															
4	32	Identification												Flags		Fragment Offset																	
8	64	Time To Live				Protocol				Header Checksum																							
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

On voit bien que la longueur d'une adresse IP est de 32 Bits de longueur (ou 4 bytes)
Si l'on prend l'exemple de l'adresse 192.168.1.254 :



Chaque numéro représente 8 bits.

Il est possible de convertir ces chiffres de décimal en binaire ce qui donnerait les chiffres suivants :



Avant de voir comment fonctionnent les binaire revoyons comment fonctionne le décimal :

Il y a 10 chiffres dans le décimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ont dis aussi que c'est un comptage en base 10.

Il y a 16 chiffres en Hexadécimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

donc si l'on fais par exemple $D * 16$ ($D = 13$) se sera égal à 208 en décimal

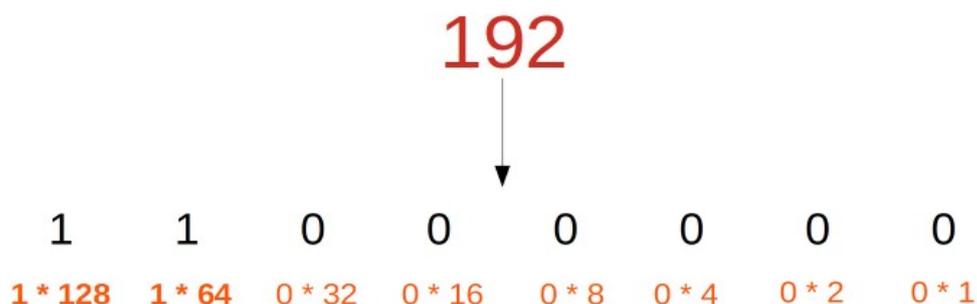
Si l'on fais $C * 256$ ($C = 12$) se sera égal à 3072 en décimal

si l'on additionne $C + D$ se sera égal à $208 + 12 = 220$

Voyons à présent comment fonctionne le Binaire.

Pour convertir 1100000 en décimal il faut utiliser la méthode suivante :

Multiplier par 2 à chaque fois en partant de droite et multiplier au nombre correspondant ainsi :



Pour 168 cela donnera :

$$\begin{array}{cccccccc} & & & & 168 & & & \\ & & & & \downarrow & & & \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 * 128 & 0 * 64 & 1 * 32 & 0 * 16 & 1 * 8 & 0 * 4 & 0 * 2 & 0 * 1 \\ 128 & + & 32 & + & 8 & = & 168 & \end{array}$$

Pour 1 cela donnera :

$$\begin{array}{cccccccc} & & & & 1 & & & \\ & & & & \downarrow & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 * 128 & 0 * 64 & 0 * 32 & 0 * 16 & 0 * 8 & 0 * 4 & 0 * 2 & 1 * 1 \end{array}$$

Pour 254 cela sera égal à :

$$\begin{array}{cccccccc} & & & & 254 & & & \\ & & & & \downarrow & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 * 128 & 1 * 64 & 1 * 32 & 1 * 16 & 1 * 8 & 1 * 4 & 1 * 2 & 0 * 1 \\ 128 + 64 + 32 + 16 + 8 + 4 + 2 & = & 254 & \end{array}$$

Essayons de convertir 10001111 en décimal se sera égal à :

Il faut multiplier par deux la position de chaque chiffre pour qu'il soit converti en décimal :

$$\begin{array}{cccccccc} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}$$

Une fois que l'on a fais cela il faut juste additionner les numéro sur lesquelles il y a un 1 ce qui donne :

$$\begin{array}{cccccccc} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 128 & & + & & 8 & + & 4 & + & 2 & + & 1 \end{array}$$

$$= 143$$

Un autre exemple on veut convertir 01110110 en décimal.

Il faut d'abord multiplier par deux la position de chaque chiffre pour qu'il soit converti en décimal :

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	0

On peut à présent additionner les numéros sur lesquelles il y a un 1 se qui donne :

128	64	32	16	8	4	2	1		
0	1	1	1	0	1	1	0		
	64	+	32	+	16	+	4	+	2

= 118

Essayons un 3ème exemple pour 11101100.

On multiplie par deux à chaque numéro et on additionne le résultat ce qui donne :

1	1	1	0	1	1	0	0	
128	+	64	+	32	+	8	+	4

= 236

A présent essayons de convertir du décimal vers le binaire :

Il faut ajouter déjà les 8 chiffres possible en binaire en multipliant à chaque fois par 2 en partant de la droite puis soustraire à chaque fois jusqu'à réduire à 0 le chiffre décimal donc pour 221 ça fera :

128 64 32 16 8 4 2 1

221 93 28 12 4
 -128 -64 -16 -8 -4
 =93 =28 =12 =4 =0

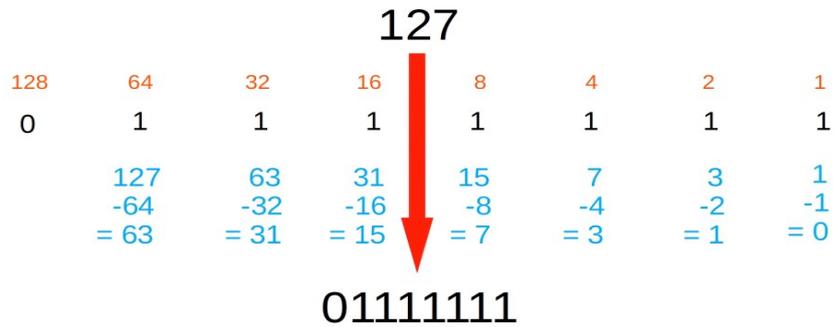
On met un « 1 » à chaque endroit sur lequel il y a eu une soustraction se qui fera :

221							
128	64	32	16	8	4	2	1
1	1	0	1	1	1	0	0
221	93		28	12	4		
-128	-64		-16	-8	-4		
= 93	= 28		= 12	= 4	= 0		

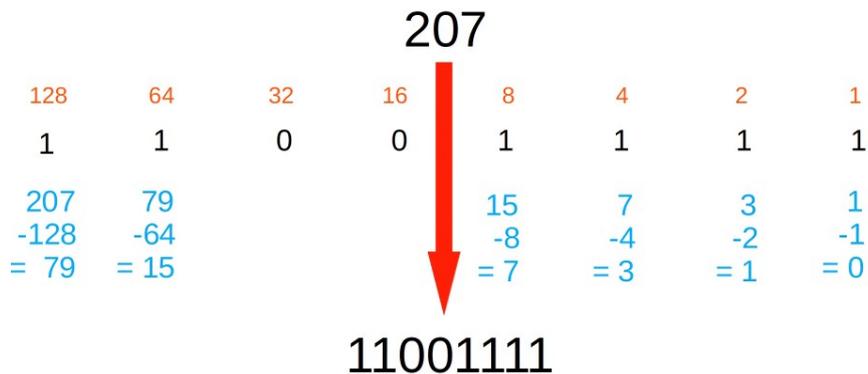
11011100

Un autre exemple pour convertir 127 en binaire.

Il faut déjà multiplier par deux à chaque chiffre décimal, puis ajouter le chiffre « 1 » à chaque endroit où il y a eu une soustraction se qui sera égal à :

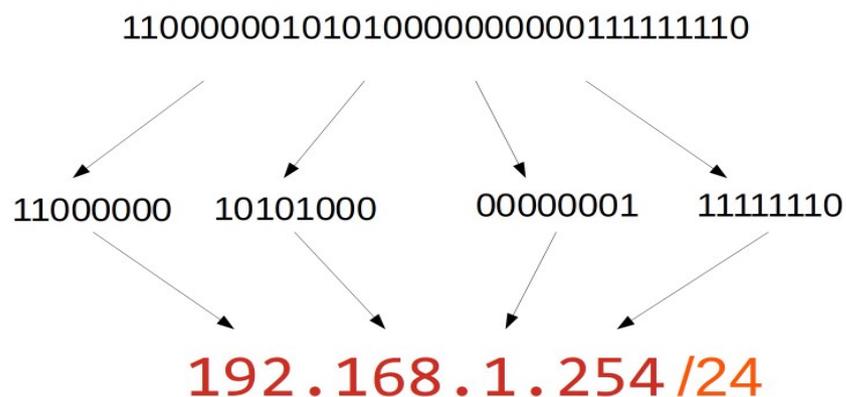


Faisons un dernière exemple pour convertir 207 en binaire :



Il est à noter que 00000000 est égal à 0 en décimal et que 11111111 est égal à 255 en décimal.

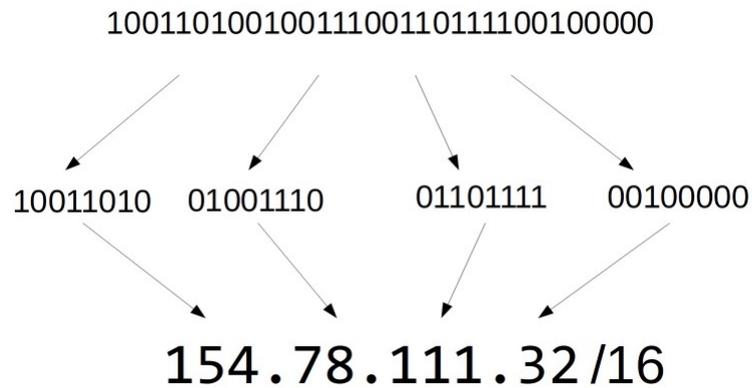
A présent que nous avons vu comment convertir des chiffres voyons comment convertir des adresses IP. Par exemple pour l'adresse : 110000001010100000000011111110 il faut déjà séparer l'adresse en 4 partie de 8 puis la convertir se qui fera :



Ici le /24 représente les 24 bits qui font partie du réseau et qui sont fixes les 8 bits restants (24 + 8 = 32) correspondent à l'adresse de l'hôte.

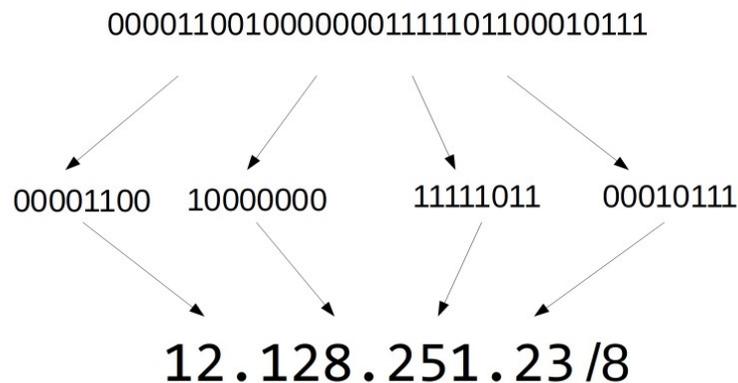
Donc ici 254 est l'adresse d'hôte du réseau.

Pour convertir l'adresse 10011010010011100110111100100000 on sépare déjà l'adresse en 4 partie de 8 chiffres puis on convertie en décimal se qui fera :



Ici le /16 indique que les 16 premiers chiffres binaires représente l'adresse du réseau fixe. Donc ici l'adresse réseau est 154.78

Un dernier exemple pour convertir 00001100100000001111101100010111 cela donne :



Ou 12 est l'adresse réseau puisque ici il y a un /8

Voyons à présent les différentes classes d'adresses IP qui sont définis en fonction du premier octet :

Classe	Premier Octet	Premier Octet avec un classement numérique
A	0xxxxxxx	0-127
B	10xxxxxx	128-191
C	110xxxxx	192-223
D	1110xxxx	224-239
E	1111xxxx	240-255

Les classes principales sont les classes A, B, et C

Les adresses des classes D sont utilisés pour les adresses Multicast

Les adresses des classes E sont réservés pour une utilisation expérimental

La fin de l'adresse de classe A est terminé par 126 et non pas 127 voici pourquoi :

Il y a une adresse de loopback qui à pour classement : 127.0.0.0 – 127.255.255.255

Ces adresses sont utilisés pour tester sa connectivité réseau sur l'appareil local.

Par exemple si l'on ping sur les adresses 127.0.0.1 et 127.23.68.241 on obtient les résultats suivants qui sont égal à 0 puisque le ping est fais sur son propre appareil.

```

C:\Users\user>ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>ping 127.23.68.241
Pinging 127.23.68.241 with 32 bytes of data:
Reply from 127.23.68.241: bytes=32 time<1ms TTL=128

Ping statistics for 127.23.68.241:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

On peut classer le réseau de cette façon :

Classes	Premier Octet	Premier Octet avec classement numérique	Longueur du préfixe
A	0xxxxxxx	0-127	/8
B	10xxxxxx	128-191	/16
C	110xxxxx	192-223	/24

Voici une capture d'écran Wikipédia pour mieux comprendre comment sont répartis les classes :

Class	Leading bits	Size of network number bit field	Size of rest bit field	Number of networks	Addresses per network
Class A	0	8	24	128 (2 ⁷)	16,777,216 (2 ²⁴)
Class B	10	16	16	16,384 (2 ¹⁴)	65,536 (2 ¹⁶)
Class C	110	24	8	2,097,152 (2 ²¹)	256 (2 ⁸)

Donc les classes sont répartis avec les masques suivants :

- Classes A : /8 255.0.0.0
- Classes B : /16 255.255.0.0
- Classes C : /24 255.255.255.0

Si la portion d'hôte de l'adresse est 0 cela signifie qu'il s'agit de l'adresse réseau par exemple 192.168.1.0/24 cette adresse ne peut pas utiliser pour un hôte la première adresse est donc 192.168.1.1

Si la portion de l'adresse est 1 cela signifie qu'il s'agit de l'adresse de Broadcast, cette adresse ne peut pas être assigné à un hôte, ici c'est 192.168.1.254