

Tableau des masques de sous-réseau valides

Masque décimal	Masque binaire	CIDR	Bits hôtes	Hôtes possibles (2 ⁿ - 2)
255.0.0.0	11111111.00000000.00000000.00000000	/8	24	16 777 214
255.128.0.0	11111111.10000000.00000000.00000000	/9	23	8 388 606
255.192.0.0	11111111.11000000.00000000.00000000	/10	22	4 194 302
255.224.0.0	11111111.11100000.00000000.00000000	/11	21	2 097 150
255.240.0.0	11111111.11110000.00000000.00000000	/12	20	1 048 574
255.248.0.0	11111111.11111000.00000000.00000000	/13	19	524 286
255.252.0.0	11111111.11111100.00000000.00000000	/14	18	262 142
255.254.0.0	11111111.11111110.00000000.00000000	/15	17	131 070
255.255.0.0	11111111.11111111.00000000.00000000	/16	16	65 534
255.255.128.0	11111111.11111111.10000000.00000000	/17	15	32 766
255.255.192.0	11111111.11111111.11000000.00000000	/18	14	16 382
255.255.224.0	11111111.11111111.11100000.00000000	/19	13	8 190
255.255.240.0	11111111.11111111.11110000.00000000	/20	12	4 094
255.255.248.0	11111111.11111111.11111000.00000000	/21	11	2 046
255.255.252.0	11111111.11111111.11111100.00000000	/22	10	1 022
255.255.254.0	11111111.11111111.11111110.00000000	/23	9	510
255.255.255.0	11111111.11111111.11111111.00000000	/24	8	254
255.255.255.128	11111111.11111111.11111111.10000000	/25	7	126
255.255.255.192	11111111.11111111.11111111.11000000	/26	6	62
255.255.255.224	11111111.11111111.11111111.11100000	/27	5	30
255.255.255.240	11111111.11111111.11111111.11110000	/28	4	14
255.255.255.248	11111111.11111111.11111111.11111000	/29	3	6
255.255.255.252	11111111.11111111.11111111.11111100	/30	2	2

Taille du bloc	Masque (dernier octet)	Comment obtenir les multiples	Multiples (début de bloc)	Plages couvertes
8	248	On fait : $8 \times 0, 8 \times 1, 8 \times 2, 8 \times 3...$	0, 8, 16, 24, ..., 248	0-7, 8-15, 16-23, ..., 248-255
16	240	On fait : $16 \times 0, 16 \times 1, 16 \times 2...$	0, 16, 32, 48, ..., 240	0-15, 16-31, 32-47, ..., 240-255
32	224	On fait : $32 \times 0, 32 \times 1, 32 \times 2...$	0, 32, 64, 96, 128, 160, 192, 224	0-31, 32-63, 64-95, ..., 224-255
64	192	On fait : $64 \times 0, 64 \times 1, 64 \times 2...$	0, 64, 128, 192	0-63, 64-127, 128-191, 192-255
128	128	On fait : $128 \times 0, 128 \times 1$	0, 128	0-127, 128-255
256	0	On fait : 256×0	0	0-255

Résumé : Nombre d'hôtes en fonction du masque

- Si le masque est de la forme 255.255.255.BBBBBBBB

- La partie hôte (machine sur le réseau) est notée **h**
- La partie sous réseau est notée **s**
- On dit parfois qu'on emprunte des bits à la partie hôte

Bits d'ID de sous-réseau	Bits d'ID d'hôte	Nombre de sous-réseaux	Nombre d'hôtes	Configuration binaire
0	8	1	254	hhhhhhhh
1	7	2	126	shhhhhhh
2	6	4	62	sshhhhhh
3	5	8	30	sssghhhh
4	4	16	14	ssssghhh
5	3	32	6	sssssghhh
6	2	64	2	ssssssghh

A — LES BASES (toujours vraies)

1) Rôle d'un masque

Une adresse IP = 4 octets :

A . B . C . D

Le masque sépare l'adresse en :

- **bits réseau** (1)
- **bits hôtes** (0)

En binaire :

1 = réseau

0 = hôte

Toute la logique du subnetting vient de là.

2) Lire un masque simple

Exemple :

255.255.255.0

En binaire :

11111111 . 11111111 . 11111111 . 00000000

On compte les 1 → **24**

C'est un **/24**

3) Exemple simple : /24

IP : 128.107.0.55

Masque : 255.255.255.0 (/24)

Découpage :

[128] [107] [0] [55]

R R R H

Donc :

- Adresse réseau → .0
- Premier hôte → .1
- Dernier hôte → .254
- Broadcast → .255

Automatique.

B — MÉTHODE DES BLOCS (UNIQUEMENT pour masques valides)

4) Quand un masque n'est pas 255 ou 0

Exemple :

255.255.240.0

En binaire :

11111111 . 11111111 . 11110000 . 00000000

On compte les bits à 1 → **/20**

5) Taille du bloc

On regarde l'octet du masque qui n'est **ni 255 ni 0**.

Ici : **240**

Formule :

Taille du bloc = 256 - valeur du masque

Donc, **dans ce cas précis** :

256 - 240 = 16

Dans CET exemple, le bloc fait 16 adresses.

Le réseau avance par pas de **16**.

6) Début de blocs (pour un bloc de 16)

Comme le bloc = **16**, les débuts sont :

0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240

Ce sont les **débuts de blocs pour un bloc de 16**.

7) Plages couvertes (pour un bloc de 16)

Chaque bloc couvre **16 adresses** (car bloc = 16).

- 0 → 0-15
- 16 → 16-31
- 32 → 32-47
- 48 → 48-63
- 64 → 64-79
- 80 → 80-95
- 96 → 96-111
- 112 → 112-127
- 128 → 128-143
- 144 → 144-159
- 160 → 160-175
- 176 → 176-191
- 192 → 192-207
- 208 → 208-223
- 224 → 224-239
- 240 → 240-255

Ceci n'est vrai QUE parce que le bloc = 16 dans cet exemple.

C — RÈGLE GÉNÉRALE

Chaque bloc couvre exactement SA taille.

- Bloc **8** → 8 adresses
- Bloc **16** → 16 adresses
- Bloc **32** → 32 adresses
- Bloc **64** → 64 adresses
- Bloc **128** → 128 adresses
- Bloc **256** → 256 adresses

C'est la règle universelle.

D — APPLICATION À TON EXEMPLE : 128.107.0.55 /20

3^e octet = **0**

On cherche dans quel bloc il tombe :

0-15 → oui

Donc :

- Adresse réseau → **128.107.0.0**
 - Broadcast → **128.107.15.255**
-

E — HÔTES

- Premier hôte → réseau + 1 → **128.107.0.1**
- Dernier hôte → broadcast - 1 → **128.107.15.254**

F — RÉSULTAT FINAL DU /20

Élément	Valeur
Adresse réseau	128.107.0.0
Premier hôte	128.107.0.1
Dernier hôte	128.107.15.254
Broadcast	128.107.15.255

G — MASQUES VALIDES vs NON VALIDES

Masques valides (les 9 valeurs officielles)

255
254
252
248
240
224
192
128
0

Si un octet du masque n'est PAS dans cette liste →

masque NON valide

Exemples non valides :

14, 133, 156, 55, 102

255.254.14.0

133.156.55.102

H — MASQUES NON VALIDES → MÉTHODE AND OBLIGATOIRE

Pourquoi ?

Parce que :

- 256 – masque
- blocs
- incréments
- CIDR
- sous-réseaux

NE FONCTIONNENT PAS si le masque n'est pas contigu.

Donc → **AND binaire obligatoire**

Exemple : 24 AND 14

24 → 00011000

14 → 00001110

AND → 00001000 → **8**

Aucune soustraction.

Aucun bloc.

Aucun CIDR.

I — RÉSUMÉ FINAL

Si masque **valide** → méthode des blocs

- 256 – masque
- taille du bloc
- débuts de blocs
- réseau / broadcast / hôtes

Si masque **non valide** → méthode AND

- conversion binaire
- AND
- reconversion

CIDR = nombre de bits à 1

(uniquement pour masques valides)

Exercices corrigés vus en cours :

Cas 1

IP : 184.18.24.96

Masque : 255.254.14.0

On travaille octet par octet.

Octet 1 : 184 ET 255

184 = 10111000

255 = 11111111

→ AND = 10111000

10111000 = **128 + 32 + 16 + 8 = 184**

Octet 2 : 18 ET 254

18 = 00010010

254 = 11111110

→ AND = 00010010

00010010 = **16 + 2 = 18**

Octet 3 : 24 ET 14

24 = 00011000

14 = 00001110

→ AND = 00001000

00001000 = **8**

Octet 4 : 96 ET 0

96 = 01100000

0 = 00000000

→ AND = 00000000

00000000 = **0**

Résultat 1 : 184.18.8.0

Cas 2

IP : 190.22.36.93

Masque : 133.156.55.102

Même chose : octet par octet.

Octet 1 : 190 ET 133

190 = 10111110

133 = 10000101

→ AND = 10000100

10000100 = 128 + 4 = **132**

Octet 2 : 22 ET 156

22 = 00010110

156 = 10011100

→ AND = 00010100

00010100 = 16 + 4 = **20**

Octet 3 : 36 ET 55

36 = 00100100

55 = 00110111

→ AND = 00100100

00100100 = 32 + 4 = **36**

Octet 4 : 93 ET 102

93 = 01011101

102 = 01100110

→ AND = 01000100

01000100 = 64 + 4 = **68**

Résultat 2 : 132.20.36.68

Pourquoi Cas 1 = AND ?

Masque : 255.254.14.0

→ Il contient 254 (valide)

→ mais aussi 14 (NON valide)

Donc :

pas de CIDR

pas de blocs

pas de 256 – masque

pas de sous-réseaux

AND obligatoire

Pourquoi Cas 2 = AND ?

Masque : 133.156.55.102

→ 133 = NON valide

→ 156 = NON valide

→ 55 = NON valide

→ 102 = NON valide

Donc :

pas de CIDR

pas de blocs

pas de 256 – masque

pas de sous-réseaux

AND obligatoire

Résumé ultra simple :

Cas 1 → masque non valide → AND

Cas 2 → masque non valide → AND

Les deux cas utilisent la méthode AND.

Les deux cas sont cohérents.

Les deux cas sont bien présentés.

