III. Subnetting IP et VLSM

Le schéma précédent montre que tout réseau IP possède nativement un *masque*. Et malheureusement, par abus de langage, on parlera parfois de « réseau 10.0.0.0 » pour désigner 10.0.0.0/8, de « réseau 128.0.0.0 » pour 128.0.0.0/16 ou de « réseau 192.0.0.0 » pour 192.0.0.0/24. À une époque où les masques variables (on parle également de *masques réduits*) sont légion, gardons en tête qu'***un réseau IP n'a aucune signification sans son masque associé !***

Attardons-nous un peu sur la manipulation de longueur de masques réseau, communément appelée *VLSM* (*Variable Length Subnet Mask*) parce que la maîtrise de ces opérations est au cœur du sujet qui nous intéresse. Énonçons le 1er principe de base des manipulations VLSM.

> **1er principe : étant donné un bloc IP, le masque réduit définit le nombre de réseaux disponibles ainsi que le nombre maximal de stations qu'il est possible d'adresser dans chacun de ces réseaux.**

On peut également formuler ce principe de la façon suivante : le nombre maximal de stations à adresser impose la longueur du masque réduit.

À nos calculettes donc, et prenons pour exemple le bloc 128.16.0.0/19…

Le schéma des classes d'adresses indique que ce bloc appartient à l'espace d'adressage de classe B (masque natif /16). Le /19 indique que trois bits supplémentaires sont utilisés pour adresser les réseaux. Les bits restants de l'adresse IP, soit 13 bits, permettront d'adresser les stations sur chacun de ces réseaux. En décomposant le 3e octet en binaire, les réseaux résultants seront de la forme 128.16.nnn00000.0, où n peut prendre la valeur 0 ou 1.

Ainsi, 128.16.0.0/19 définit huit sous-réseaux qui sont :

- 128.16.**000**00000.0 soit 128.16.0.0/19 ;
- 128.16.**001**00000.0 soit 128.16.32.0/19 ;
- 128.16.**010**00000.0 soit 128.16.64.0/19 ;
- 128.16.**011**00000.0 soit 128.16.96.0/19 ;
- 128.16.**100**00000.0 soit 128.16.128.0/19 ;
- 128.16.**101**00000.0 soit 128.16.160.0/19 ;
- 128.16.**110**00000.0 soit 128.16.192.0/19 ;
- 128.16.**111**00000.0 soit 128.16.224.0/19.

Quant à la partie *host* de ces huit blocs, 2^{13} adresses IP, soit 8192, seront disponibles par sous-réseau. En revanche, les « extrémités » de ces sous-blocs sont réservées au réseau logique (partie *host* à 0) et au *local broadcast ou adresse de diffusion du sous-réseau* (partie *host* à 1). Donc au total, il y aurait possibilité d'adresser 8190(8192-2=8190) stations sur chacun de ces sous-réseaux /19.

Là où les choses commencent à se compliquer un peu, c'est lorsqu'on doit subdiviser un bloc IP en plusieurs sous-blocs de largeur variable. Le 2e principe du VLSM va nous être d'une grande utilité.

> **2e principe du VLSM : lorsqu'on est amenés à subdiviser un bloc IP en plusieurs sous-réseaux sur un point de routage, on alloue les sous-blocs du plus grand au plus petit et on fait en sorte que les sous-blocs soient contigus.**

Illustrons ceci avec le scénario suivant :

- un routeur possède trois interfaces pour connecter trois réseaux IP N1, N2 et N3. L'administrateur réseau impose les conditions suivantes :
  - capacité d'adressage de N1, 40 hosts,
  - capacité d'adressage de N2, 80 hosts,
  - capacité d'adressage de N3, 140 hosts,
  - utilisation *au mieux* du bloc 128.203.0.0/20 ;
- la première étape consiste à trouver le nombre de *hosts bits*, ce qui correspond à la puissance 2 immédiatement supérieure du nombre de stations :
  - N1 : 40 hosts donc 6 bits ($2^5 < 40 < 2^6$),
  - N2 : 80 hosts donc 7 bits ($2^6 < 80 < 2^7$),
  - N3 : 140 hosts donc 8 bits ($2^7 < 140 < 2^8$) ;
- par conséquent, nous pouvons maintenant dimensionner les masques puisque nous connaissons le nombre de bits réservés aux stations (le bloc alloué 128.203.0.0/20 utilise déjà les quatre premiers bits du octet) :
  - masque N1, 255.255.255.192 (/26),
  - masque N2, 255.255.255.128 (/25),
  - masque N3, 255.255.255.0 (/24) ;
- On applique maintenant le principe énoncé plus haut :
  - 1er bloc, le plus grand, N3. Masque 255.255.255.0 (/24) soit 128.203.0.0/24. Cet espace d'adressage s'étend de 128.203.0.0 à 128.203.0.255 (253 stations),
  - vient ensuite le 2e bloc, N2. Masque 255.255.255.128 (/25). On utilise le sous-bloc contigu à N1. Ce sera donc 128.203.1.0/25. Cet espace d'adressage s'étend de 128.203.1.0 à 128.203.1.127 (126 stations),
  - enfin, le plus petit sous-réseau, N1. Masque 255.255.255.192 /(26). On utilise le sous-bloc contigu à N2, soit 128.203.1.128/26 qui représente l'espace d'adressage 128.203.201.128 à 128.203.1.191 (62 stations).

N3

128.203.1.128

En utilisant cet algorithme, l'adressage IP est optimisé au maximum, impossible de faire mieux. Si vous êtes férus en programmation, c'est un très joli challenge… Je l'avais codé en C dans les années 2000 pour un gros client qui l'utilise toujours ! C'est la façon dont les ISP gèrent leurs espaces d'adressage IP. Le deuxième intérêt de cette méthode est qu'il prépare le travail pour le *supernetting* comme je vais vous le montrer dans la partie suivante.

IV. Supernetting IP

Concrètement, le *supernetting IP* est l'opération inverse du *subnetting*. La littérature le désigne par CIDR (Classless Inter-Domain Routing). Étant donné plusieurs réseaux IP, comment représenter cet espace d'adressage à l'aide d'un couple (*réseau_logique, masque*) ? En somme, comment agréger toutes ces routes IP ? Outre une meilleure gestion des espaces d'adressage dans un système autonome, le principal atout du supernetting est la réduction des tables de routage.

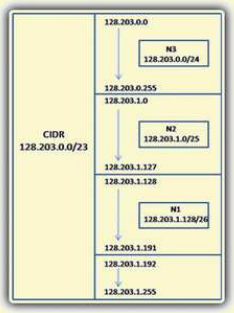
En effet, en reprenant notre scénario précédent, supposons que le routeur qui connecte les réseaux 128.203.0.0/24, 128.203.1.0/25 et 128.203.1.128/26 ait à propager sa table de routage locale (par le biais bien sûr d'un protocole qui propage également les masques de sous-réseau tels OSPF, RIPv2, BGP, etc.). Si nous pouvons trouver un couple judicieux (*reseau_logique, masque*), le routeur enverra une seule route IP au lieu de trois. Autant de ressources mémoires et CPU économisées !

Plus précisément, revenons à la forme binaire des 3e et 4e octets des masques réseaux attachés à ce routeur :

- N2 (/25) 255.255.11111111.10000000 ;
- N3 (/24) 255.255.11111111.00000000.

Il apparaît clairement qu'il est possible d'inclure l'ensemble des réseaux N1, N2 et N3 en utilisant le masque 255.255.11111110.00000000 soit /23. Le meilleur *supernet CIDR* qu'on puisse trouver pour annoncer ces trois routes IP est 128.203.0.0/23. Cet espace d'adresses s'étend depuis 128.203.0.0 jusqu'à 128.203.1.255. D'autre part, d'après ce qui précède, il est évident que **le supernet CIDR censé annoncer plusieurs subnets d'un même réseau de classe A, B ou C (on parle du *major network*) aura toujours un masque inférieur ou égal au plus petit des masques des subnets.**

Le schéma suivant illustre l'imbrication des sous-blocs IP que nous avons utilisés dans ce scénario :



On constate ici que les subnets IP N1, N2 et N3 couvrent quasiment tout le CIDR. Il reste quelques adresses disponibles (le sous-bloc 128.203.1.192/26). Notre CIDR « gaspille » une soixantaine d'adresses IP, ce qui est somme toute un très bon résultat. Ces adresses seront parfois utiles pour déployer de nouveaux réseaux en cas de besoin. Encore une fois, et sans rentrer dans la démonstration, l'algorithme que j'ai illustré ici avec ce scénario est la seule méthode qui permette d'optimiser au mieux un bloc IP éclaté en plusieurs sous-réseaux. Parce que par construction, au fil du découpage des sous-blocs, on déplace le dernier bit du masque réseau « vers la droite »…

Dans le cas général, il faut revenir à l'écriture binaire pour trouver le supernet le plus fin qui permette d'annoncer plusieurs réseaux issus du même *major network*. En revanche, il est très dangereux de modifier des tables de routage par introduction de CIDR sur un gros réseau sans étudier avec minutie la façon dont sont distribués les sous-blocs IP. Parce qu'il y a toujours le risque qu'un sous-bloc soit utilisé quelque part en aval du routeur qui annonce le CIDR… Ce sera notamment le cas de sous-blocs qui seraient « masqués » derrière des routes par défaut par exemple. Les paquets à destination de ces réseaux suivraient immanquablement le chemin du CIDR (cas de *misrouting*). Dans ce cas de figure, il est parfois judicieux d'optimiser les tables de routage en utilisant deux, voire trois CIDR lorsque les sous-blocs manquants ont été identifiés. Les exercices 3, 4 et 5 que je propose un peu plus loin dans ce document illustrent ce type de situation.

Dans certaines architectures, on évitera l'emploi des CIDR parce qu'on préférera propager les réseaux de façon explicite. Les tables de routage sont alors scrutées en permanence par des scripts qui remontent des alarmes lorsque certains réseaux disparaissent (réseaux de *trading* et *datacenters* par exemple ou bien encore *loopbacks* de routeurs).

Un autre inconvénient survient également lorsqu'un réseau contenu dans un CIDR n'est plus joignable. Le CIDR étant toujours annoncé, les stations tenteront d'émettre vers le réseau et c'est le routeur source du CIDR qui enverra un *ICMP (Internet Control Message Protocol) Destination Unreachable*, au lieu de la *gateway* locale de la station.