

Réseaux - Cours 5

IP : Routage, Sous-Réseaux, CIDR

Cyril Pain-Barre

IUT Informatique Aix-en-Provence

version du 22/2/2013

Routage IP

Algorithme de routage

- le logiciel IP de **A** doit envoyer un datagramme à (l'adresse IP de) **B**, situé quelque part dans l'inter-réseau

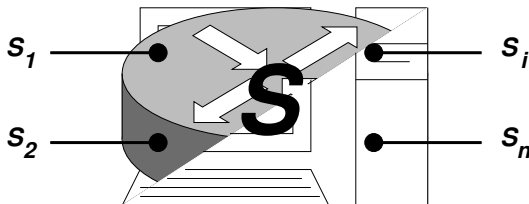


- routage = prise de décision pour l'envoi
- question : la destination appartient-elle au même réseau ?
 - oui : la remise est **directe**.
A peut envoyer directement le datagramme à **B**, en utilisant le service d'envoi de leur réseau
 - non : la remise est **indirecte**.
A ne peut qu'envoyer le datagramme à un routeur. À son tour, le routeur devra appliquer le même algorithme

dans ce cas, le choix du routeur est prépondérant

Test de l'appartenance au même réseau : version 1

- une station (ou un routeur) S dispose d'une ou plusieurs interfaces, chacune avec une adresse IP S_i



- S doit envoyer un datagramme à une IP de destination D
- pour savoir si D appartient à un réseau connecté à S :
 - de D et de sa classe, en déduire l'adresse du réseau de D , qu'on notera $R(D)$
 - pour chaque adresse IP S_i de S :
 - extraire son adresse de réseau $R(S_i)$
 - si $R(S_i) = R(D)$ alors S et D appartiennent au même réseau

Test de l'appartenance au même réseau : exemple

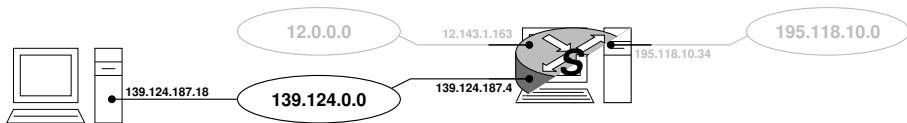
- soient S avec 3 interfaces (S_1 , S_2 et S_3) et une destination $D_1 = 139.124.187.18$:



- $S_1 = 12.143.1.163$
 - $S_2 = 139.124.187.4$
 - $S_3 = 195.118.10.34$
- l'adresse de D_1 est de classe B $\implies R(D_1) = 139.124.0.0$
 - puisque $R(D_1) = R(S_2)$, alors D_1 et S appartiennent à un même réseau
 - S peut envoyer directement un datagramme à D_1 en utilisant l'interface (et le réseau) associée à S_2

Test de l'appartenance au même réseau : exemple

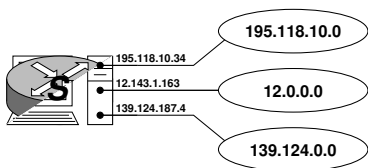
- soient S avec 3 interfaces (S_1 , S_2 et S_3) et une destination $D_1 = 139.124.187.18$:



- $S_1 = 12.143.1.163$ (classe A) $\implies R(S_1) = 12.0.0.0$
- $S_2 = 139.124.187.4$ (classe B) $\implies R(S_2) = 139.124.0.0$
- $S_3 = 195.118.10.34$ (classe C) $\implies R(S_3) = 195.118.10.0$
- l'adresse de D_1 est de classe B $\implies R(D_1) = 139.124.0.0$
- puisque $R(D_1) = R(S_2)$, alors D_1 et S appartiennent à un même réseau
- S peut envoyer directement un datagramme à D_1 en utilisant l'interface (et le réseau) associée à S_2

Test de l'appartenance au même réseau : exemple 2

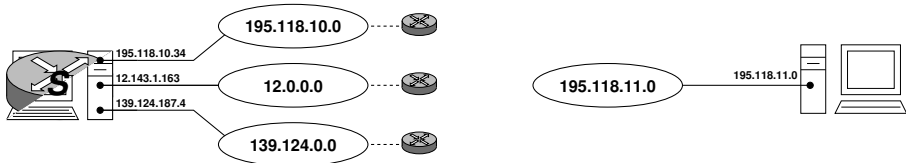
- le même S (avec 3 interfaces) doit envoyer à une destination $D_2 = 195.118.11.35$:



- $S_1 = 12.143.1.187$ (classe A) $\implies R(S_1) = 12.0.0.0$
- $S_2 = 139.124.187.4$ (classe B) $\implies R(S_2) = 139.124.0.0$
- $S_3 = 195.118.10.34$ (classe C) $\implies R(S_3) = 195.118.10.0$
- l'adresse de D_2 est de classe C $\implies R(D_2) = 195.118.11.0$
- $R(D_2)$ est différent de tous les $R(S_i)$, alors D_1 et S appartiennent à des réseaux différents
- pour envoyer un datagramme à D_2 , S doit passer par un routeur connecté à l'un de ses réseaux

Test de l'appartenance au même réseau : exemple 2

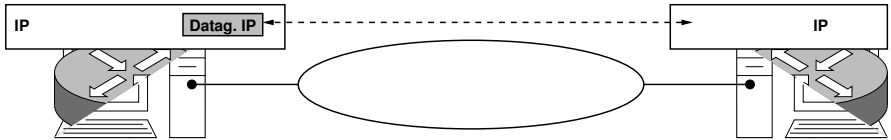
- le même S (avec 3 interfaces) doit envoyer à une destination $D_2 = 195.118.11.35$:



- $S_1 = 12.143.1.187$ (classe A) $\implies R(S_1) = 12.0.0.0$
- $S_2 = 139.124.187.4$ (classe B) $\implies R(S_2) = 139.124.0.0$
- $S_3 = 195.118.10.34$ (classe C) $\implies R(S_3) = 195.118.10.0$
- l'adresse de D_2 est de classe C $\implies R(D_2) = 195.118.11.0$
- $R(D_2)$ est différent de tous les $R(S_i)$, alors D_1 et S appartiennent à des réseaux différents
- pour envoyer un datagramme à D_2 , S doit passer par un routeur connecté à l'un de ses réseaux

La remise directe

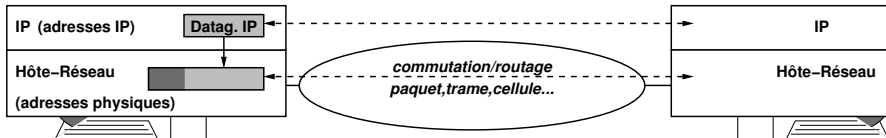
situation : une station ou un routeur doit envoyer un datagramme à un hôte situé sur le même réseau physique que lui



contrainte : l'envoi ne peut se faire qu'en utilisant des adresses physiques (MAC) et le service du réseau

La remise directe

situation : une station ou un routeur doit envoyer un datagramme à un hôte situé sur le même réseau physique que lui



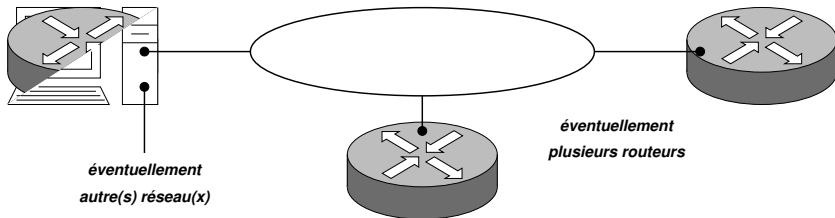
contrainte : l'envoi ne peut se faire qu'en utilisant des adresses physiques (MAC) et le service du réseau

méthode :

- 1 déterminer l'adresse physique du destinataire par **résolution d'adresse** (ARP ou autre)
- 2 utiliser le service du réseau (mode connecté ou non, fiable ou pas) pour lui transmettre le datagramme

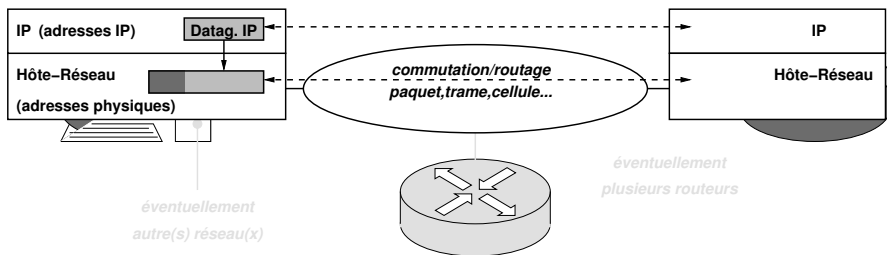
La remise indirecte

situation : une station ou un routeur ne peut pas joindre directement la destination d'un datagramme et doit le confier à un routeur



La remise indirecte

situation : une station ou un routeur ne peut pas joindre directement la destination d'un datagramme et doit le confier à un routeur



méthode :

- ① déterminer l'adresse IP du routeur à solliciter par consultation de la **table de routage**
- ② transmettre le datagramme au routeur :
 - a) déterminer son adresse physique (résolution d'adresse)
 - b) utiliser le service du réseau pour lui transmettre le datagramme

Politique de traitement d'un datagramme reçu

situation : une station/routeur reçoit un datagramme : qu'en faire ?

traitement pour un routeur

- si le datagramme est destiné au routeur, l'accepter
- sinon le routeur est utilisé comme nœud de transfert et doit relayer le datagramme (si aucune route, message ICMP renvoyé)

une station, même disposant de plusieurs interfaces, n'assure pas la fonction de nœud de transfert, à moins qu'elle soit configurée explicitement pour cela

traitement pour une station

- si le datagramme est destiné à la station, l'accepter
- sinon le datagramme est détruit et un message ICMP est renvoyé

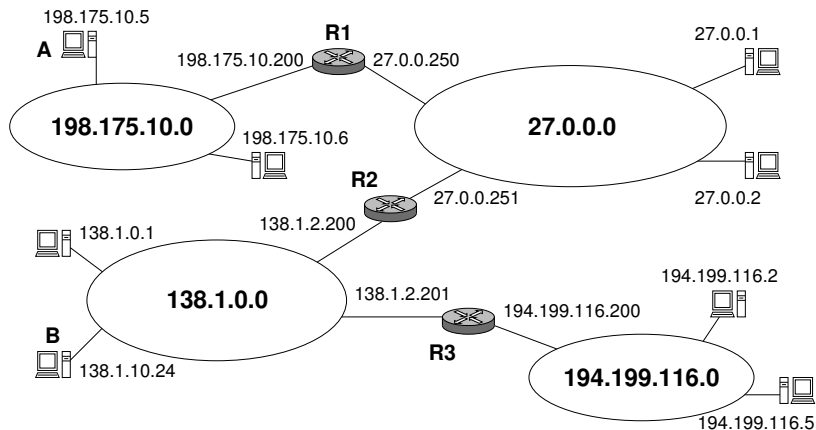
Tables de routage : version d'origine

- chaque station ou routeur dispose de sa propre table de routage
- une table contient autant d'entrées que de destinations (réseaux) connues de l'hôte
- une entrée est un couple
(adresse réseau, adresse du saut suivant (routeur))
- pour une destination donnée, la table n'indique pas le chemin à suivre, seulement le routeur à qui confier le datagramme

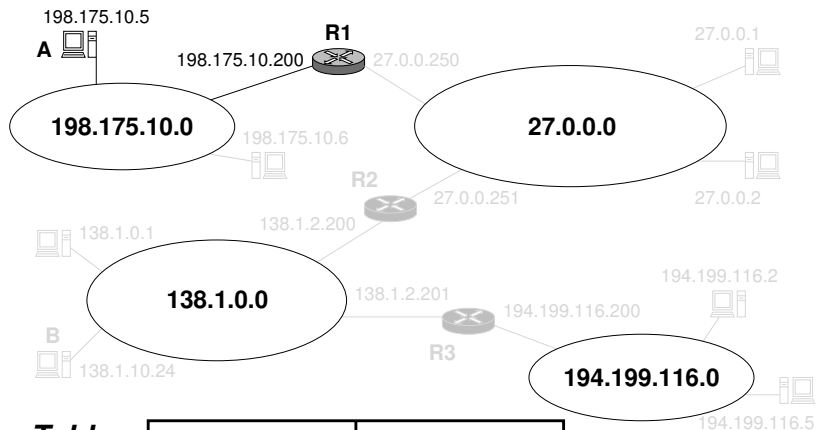
le chemin est une information répartie

- ce routeur doit être **directement accessible** (situé sur un réseau commun) et c'est son adresse dans ce réseau qui est utilisée
- une destination dont l'adresse de réseau ne figure pas dans la table est inaccessible

Tables de routage : exemple



Tables de routage : exemple



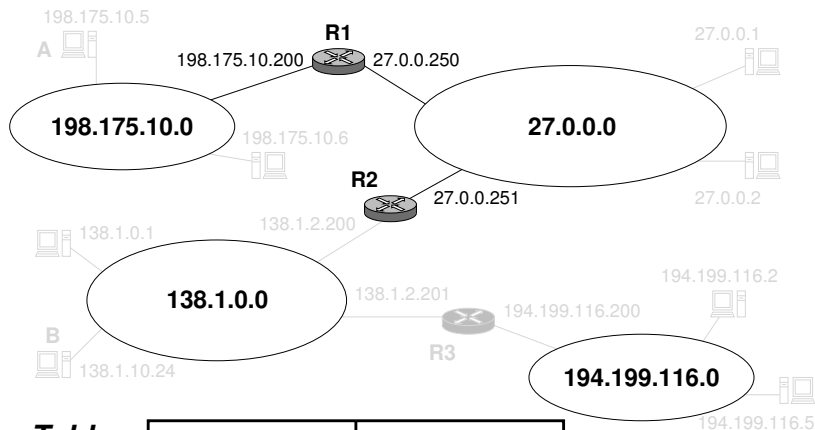
**Table
de A :**

destination	routeur
198.175.10.0	0.0.0.0
27.0.0.0	198.175.10.200
138.1.0.0	198.175.10.200
194.199.116.0	198.175.10.200

} remise directe

} remise indirecte

Tables de routage : exemple



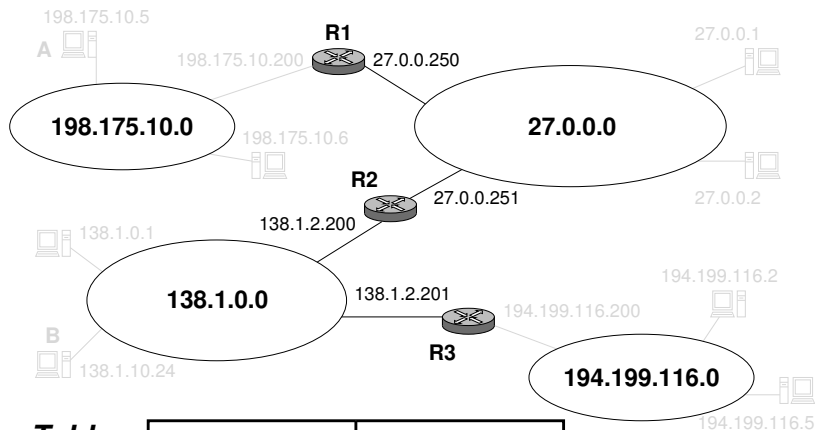
**Table
de R1 :**

destination	routeur
198.175.10.0	0.0.0.0
27.0.0.0	0.0.0.0
138.1.0.0	27.0.0.251
194.199.116.0	27.0.0.251

} remise directe

} remise indirecte

Tables de routage : exemple



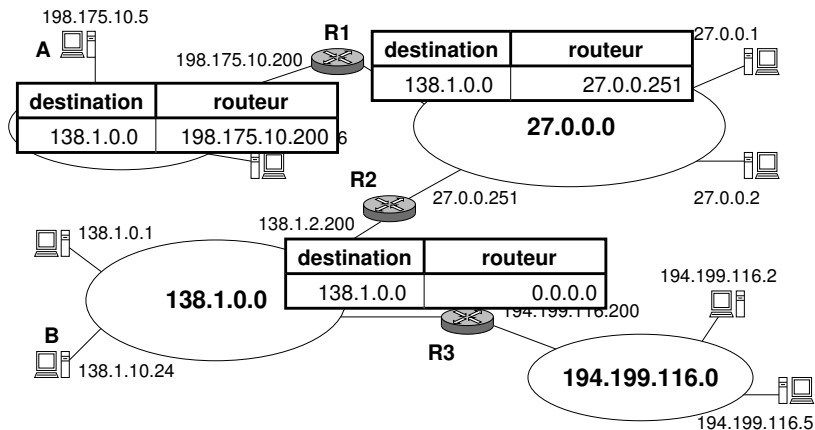
**Table
de R2 :**

destination	routeur
27.0.0.0	0.0.0.0
138.1.0.0	0.0.0.0
198.175.10.0	27.0.0.250
194.199.116.0	138.1.2.201

} remise directe

} remise indirecte

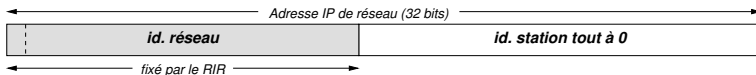
Tables de routage : exemple



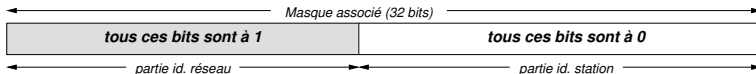
**le chemin qui mène de A à B est une information répartie :
aucun hôte ne le connaît en totalité**

Masques de sous-réseau

- une adresse IP de réseau est composée d'un *id. réseau* et d'un *id. station* :



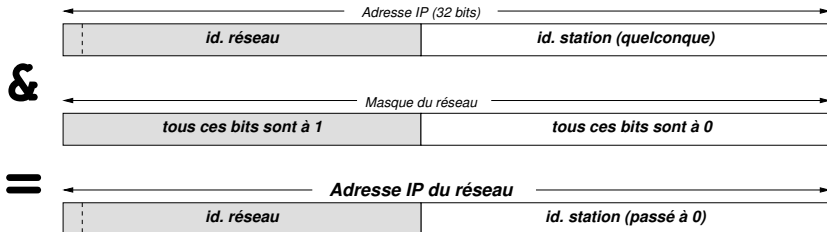
- une adresse de réseau est toujours associée à un **masque de sous-réseau** codé aussi sur 32 bits
- il indique les portions de l'adresse codant l'*id. réseau* et l'*id. station* :



- ses bits à 1 identifient la partie *id. réseau* ;
- ses bits à 0 identifient la partie *id. station*.
- En appliquant ce masque à n'importe quelle adresse de ce réseau, on retombe sur l'adresse du réseau

Application du masque de sous-réseau

- Appliquer un masque veut dire réaliser un **ET bit-à-bit** :



Utilisation des masques

- la configuration d'une interface nécessite :
 - l'adresse IP de l'hôte
 - le masque de son réseau

```
# ifconfig eth0 198.75.10.5 netmask 255.255.255.0
```

- desquels est déduite l'adresse du réseau de cet hôte (198.75.10.0)
- qui est ajoutée automatiquement comme route directe dans la table de routage :

```
# route -n
```

Table de routage IP du noyau

Destination	Passerelle	Genmask	...	Iface
198.75.10.0	0.0.0.0	255.255.255.0	...	eth0

- la table de routage **comporte aussi une colonne masque**, associé à chaque destination

- l'algorithme et les tables de routage doivent être modifiés
- dans la table, on ajoute une colonne **masque** :

destination	masque	routeur
...

- pour router un datagramme de destination D , on cherche dans la table une entrée (D_i, M_i, R_i) telle que l'application de M_i à D donne D_i
- si plusieurs entrées correspondent, on prendra celle dont le masque comporte le plus de 1 (le masque le plus précis)

- l'usage des masques fait maintenant partie du standard IP
- toutes les stations/routeurs les utilisent
- masques par défaut selon les classes :
 - 255.0.0.0 (ou /8) pour les réseaux de classe A
 - 255.255.0.0 (ou /16) pour les réseaux de classe B
 - 255.255.255.0 (ou /24) pour les réseaux de classe C

où $/i$ est la notation CIDR qui signifie que la partie identifiant le réseau se trouve sur les i premiers bits.

Route par défaut

- une destination n'est (éventuellement) accessible que si son réseau figure dans les tables de routage
- or l'inter-réseau évolue constamment (ajout/suppression de routeurs, liaisons inter-routeurs, **réseaux**)
- ce qui peut conduire à la nécessité de modifier toutes les tables de routage
- dans une majorité de cas, on peut se contenter d'utiliser une **route par défaut** qui comprend toutes les destinations non explicitement mentionnées
- particulièrement adaptée pour un hôte relié à un seul routeur
- dans la table de routage, la route par défaut vers un routeur R est simplement représentée par l'entrée :

$(0.0.0.0, 0.0.0.0, R)$

Route par défaut : exemple

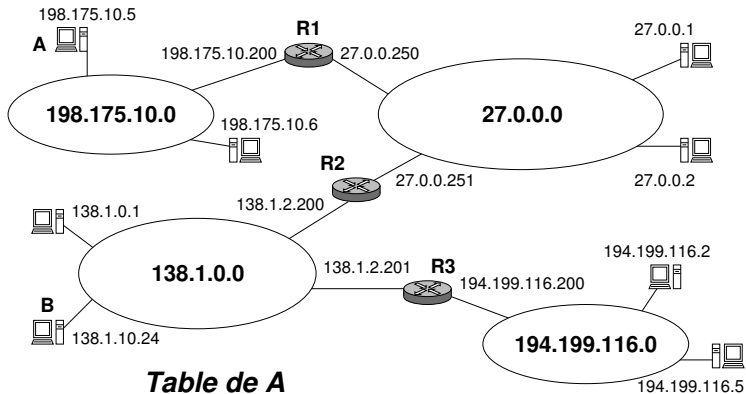


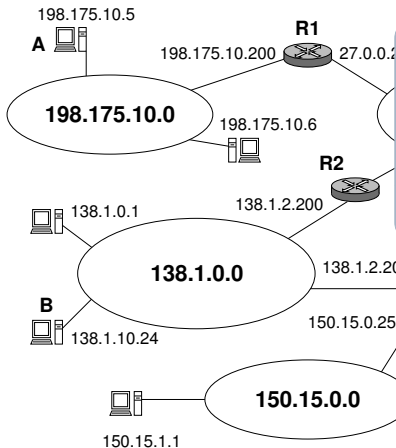
Table de A

destination	routeur
198.175.10.0	0.0.0.0
27.0.0.0	198.175.10.200
138.1.0.0	198.175.10.200
194.199.116.0	198.175.10.200



destination	routeur
198.175.10.0	0.0.0.0
défaut	198.175.10.200

Route par défaut : exemple



l'usage de ce type de route sur les routeurs doit être limité pour ne pas créer de boucle...

ce nouveau réseau est déjà pris en compte par la route par défaut

destination	routeur
198.175.10.0	0.0.0.0
défaut	198.175.10.200

Routage IP et sous-réseaux

- dans la version d'origine d'IP, une adresse IP est constituée d'un *id. réseau* et d'un *id. station* dans ce réseau
- le routage classique n'utilisait que l'*id. réseau*

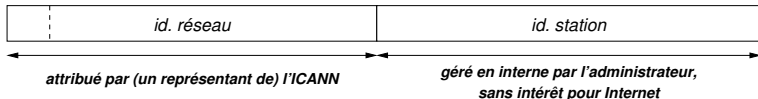
à chaque réseau physique doit être affectée une adresse de réseau unique

- or, la multiplication des réseaux pose plusieurs problèmes :
 - la gestion de toutes les adresses de réseau devient très lourde
 - les tables de routage deviennent gigantesques
 - le schéma d'adressage peut être saturé

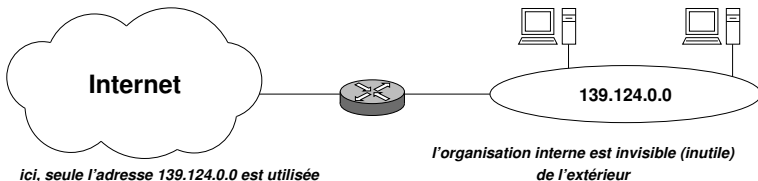
il est devenu nécessaire de réduire le nombre de réseaux à gérer, notamment en permettant à plusieurs réseaux physiques de partager la même adresse de réseau

Sous-réseaux IP : les clés de la réussite

- pour connecter un réseau à Internet, un administrateur demande une adresse de réseau (classe A, B ou C)
- l'organisation interne du réseau est à la charge de l'administrateur :
 - plan d'adressage (affectation des adresses IP). En particulier, l'administrateur utilise comme il le souhaite la partie *id. station* :

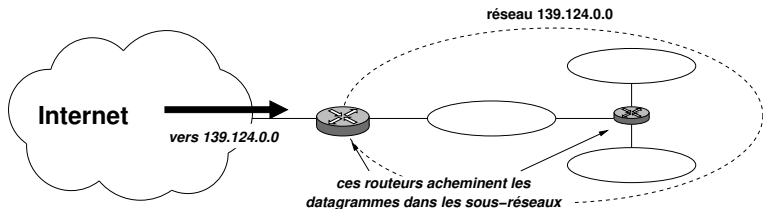


- définition des routes sur les ordinateurs et routeurs de ce réseau
- du point de vue (des autres réseaux et routeurs) d'Internet, seule l'adresse du réseau est prise en compte



Sous-réseaux IP : principe

- l'administrateur dispose d'une adresse de réseau (telle que 139.124.0.0) mais de plusieurs réseaux physiques, appelés **sous-réseaux**
- la présence de plusieurs réseaux physiques est une question interne
- les routeurs d'Internet se contentent d'acheminer les datagrammes vers "le réseau" 139.124.0.0
- à charge des routeurs internes d'acheminer les datagrammes à travers les sous-réseaux



Sous-réseaux IP : identifiant de sous-réseau

- pour distinguer les sous-réseaux, l'administrateur réserve une partie de l'*id. station*, appelée l'**identifiant de sous-réseau** :

un seul réseau (pas de sous-réseau)



avec des sous-réseaux



- les stations et routeurs internes doivent en tenir compte pour leurs décisions de routage (accessibilité directe ou indirecte de la destination)
- la taille de l'*id. sous-réseau* dépend du nombre de sous-réseaux, en tenant compte des contraintes (recommandations) suivantes :
 - l'*id. sous-réseau* tout à 0 est réservé (conflit avec l'adresse du réseau)
 - l'*id. sous-réseau* tout à 1 est réservé (conflit avec l'adresse de diffusion dirigée)

d'où : $2^n - 2$ sous-réseaux avec n bits pour l'*id. sous-réseau*

En pratique, ces recommandations ne sont pas toujours suivies car considérées obsolètes, mais nous les suivrons le plus souvent. . .

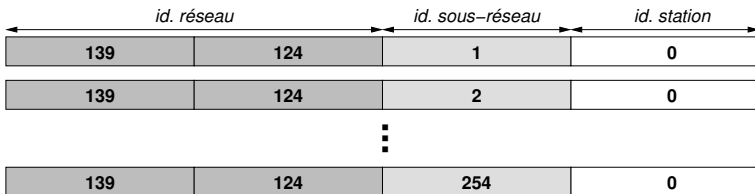
Sous-réseaux IP : exemples d'identifiant de sous-réseau

Dans le réseau 139.124.0.0 :

- si on utilise un octet pour l'*id. sous-réseau*, on peut avoir jusqu'à 254 sous-réseaux d'au plus 254 stations.

Les sous-réseaux auront pour adresses :

139.124.1.0, 139.124.2.0, ... jusqu'à 139.124.254.0



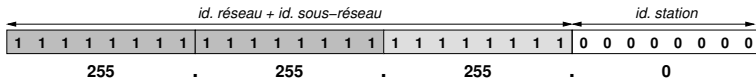
- si on n'utilise que 3 bits, on peut avoir jusqu'à 6 sous-réseaux d'au plus 8 190 stations.

Les sous-réseaux auront pour adresses :

**139.124.32.0, 139.124.64.0, 139.124.96.0, 139.124.128.0,
139.124.160.0 et 139.124.192.0**

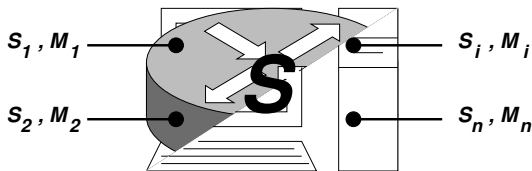
Sous-réseaux IP : masques de sous-réseaux

- en interne, pour identifier le sous-réseau, il faut prendre en compte la partie *id. sous-réseau*
- or, le schéma d'adressage en classe ne permet que d'extraire la partie *id. réseau*
- d'où l'usage généralisé des **masques de sous-réseaux** :
 - ses bits à 1 indiquent où se trouvent les parties **id. réseau** et **id. sous-réseau**
 - ses bits à 0 indiquent où se trouve la partie *id. station*
- exemple : pour le réseau 139.124.0.0 et l'*id. sous-réseau* sur un octet, le masque est 255.255.255.0 (autre notation /24) :



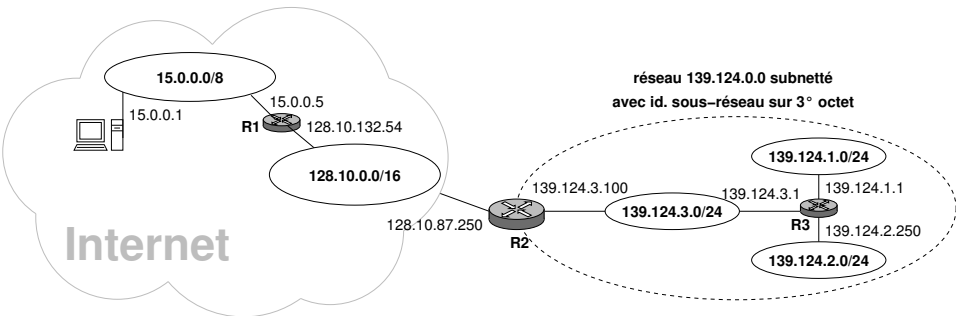
Sous-réseaux IP : test d'appartenance au sous-réseau

- à chaque interface (et adresse IP) d'une station ou routeur S d'un sous-réseau, on associe le masque du sous-réseau correspondant :

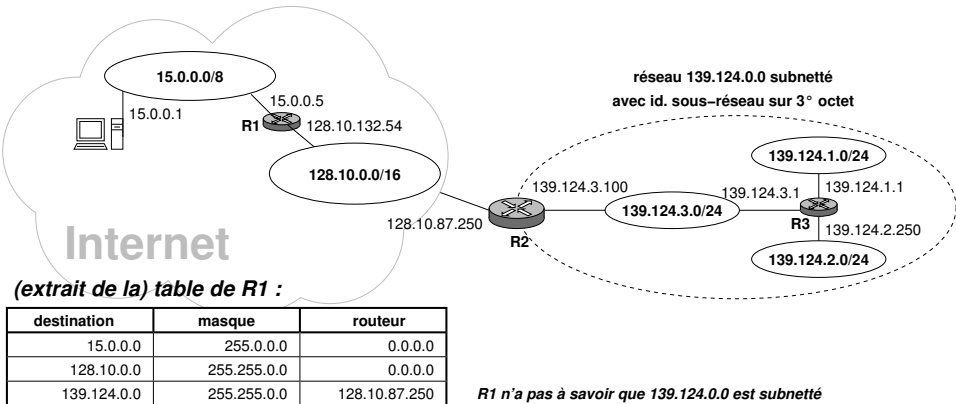


- le test d'appartenance au réseau est modifié pour tenir compte des sous-réseaux
- pour savoir si D appartient à un sous-réseau connecté à S :
 - pour chaque adresse IP S_i de S :
 - soit M_i son masque
 - extraire son adresse de réseau $R(S_i)$ en appliquant le masque M_i à S_i
 - appliquer le masque M_i à D pour obtenir D'
 - si $R(S_i) = D'$ alors S et D appartiennent au même sous-réseau

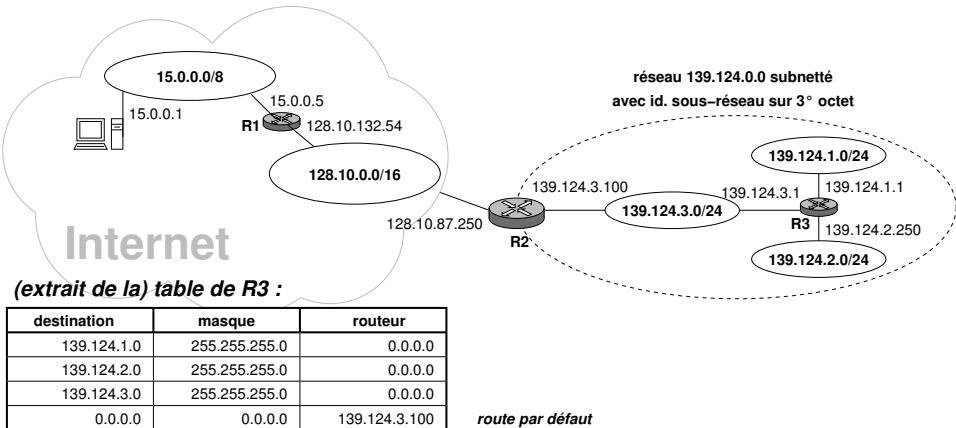
Sous-réseaux IP : exemples de tables de routage



Sous-réseaux IP : exemples de tables de routage



Sous-réseaux IP : exemples de tables de routage



- adresse de diffusion 139.124.255.255 :
 - dans tout le réseau 139.124.0.0 si non subnetté
 - dans tous les sous-réseaux de 139.124.0.0 si subnetting
- adresse 139.124.1.255 pour sous-réseau 139.124.1.0/24 :
diffusion dans le seul sous-réseau 139.124.1.0

- bien que ce ne soit pas du tout conseillé, l'identifiant sous-réseau n'est pas forcément un groupe de bits contigus
- il est possible de regrouper plusieurs entrées de la table de routage en une seule, en jouant sur le masque associé (agrégation d'adresse)
- on n'est pas limité au subnetting d'un réseau : le subnetting d'un sous-réseau est tout aussi possible
- les exemples précédents n'ont montré que le subnetting avec des sous-réseau de taille égale (la taille de la partie *id. sous-réseau* était la même pour tous les sous-réseaux). Mais on peut tout aussi bien subnetter avec des sous-réseaux de taille variable par la technique du **sous-adressage variable** (voir TP).

CIDR : *Classless Inter-Domain Routing* (RFC 1518 et 4632)

L'adressage hors classe (ou sur-adressage)

Quelques considérations historiques

- Lors de la conception d'IP et de son format d'adressage, Internet (ARPANET) ne rassemblait que les principales universités de recherche américaines, quelques entreprises et sites militaires
- En connectant les 2 000 établissements d'enseignement supérieur des Etats-Unis et de nombreuses universités étrangères, on ne devait pas dépasser 16 000 sites connectés
- Le nombre d'adresses réseau était largement suffisant : plus de 2 millions (16 383 en classe B)
- Personne ne pensait qu'Internet deviendrait un réseau public mondial
- En 1996, 100 000 réseaux étaient déjà connectés à Internet

La moitié des réseaux de classe B ne contenait pas plus de 50 hôtes !

Inconvénient des classes et du sous-adressage

En 1993, il n'y avait déjà plus d'adresse de classe B disponible, à cause du découpage en classes (A, B et C) et du sous-adressage.

À cette époque, pour une organisation donnée :

- une adresse de classe A, c'était trop
- une adresse de classe C, ce n'était pas confortable même si l'entreprise ne comptait que 50 hôtes :
 - le prix des PC rendait possible une forte croissance du parc informatique
 - plusieurs réseaux physiques allaient équiper l'organisation et la classe C n'est pas vraiment intéressante pour le sous-adressage
- en conclusion, une adresse de classe B, c'était bien mieux.

En attendant l'achèvement et le déploiement d'IPv6 avec ses adresses sur 16 octets, il a fallu trouver une solution temporaire : le **sur-adressage** ou **adressage hors-classe** (CIDR)

Une entreprise désire une adresse de classe B (notamment pour faire du sous-adressage sur le 3^e octet).

Problème : plus d'adresse de classe B disponible

Solution : lui attribuer 256 adresses de classe C

Nouveau problème : ces 256 adresses génèrent 256 entrées dans les tables de routage d'Internet pour cette seule entreprise !

Solution :

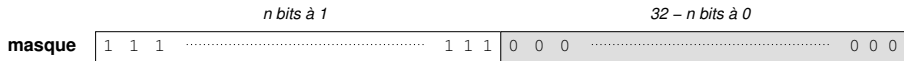
- les adresses attribuées doivent former un **bloc** d'adresses
bloc : k adresses qui se suivent, où k est une puissance de 2, et commençant à un multiple de k .

Par exemple, de 193.150.0.0 à 193.150.255.0 qui forment un bloc de $2^8 \times 2^8$ adresses

- le 3^e octet n'étant plus significatif, les 256 entrées peuvent être **agrégées** en seule entrée 193.150.0.0 de masque 255.255.0.0
- En notation CIDR, cette entrée s'écrit **193.150.0.0/16**

CIDR et le routage

- les routeurs ne peuvent plus utiliser les classes
- les tables de routage contiennent obligatoirement un masque pour chaque destination
- le masque s'écrit sous la forme $/n$, collé à la destination
- n est la longueur du préfixe : le nombre de bits à 1 du masque en partant de la gauche (pas de masques à "trous")



Exemple :

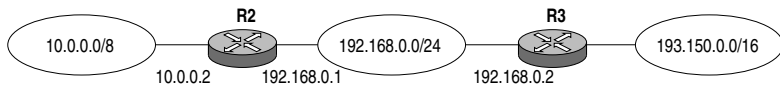


Table de R2 :

Destination/Préfixe	Routeur
10.0.0.0/8	0.0.0.0
192.168.0.0/24	0.0.0.0
193.150.0.0/16	192.168.0.2

CIDR et les entrées des tables de routage

- pour une adresse destination, plusieurs entrées peuvent correspondre
- **c'est l'entrée ayant le préfixe le plus grand qui est retenue**
- la route par défaut est l'entrée 0.0.0.0/0
- la route vers un ordinateur comme 50.51.52.53 est l'entrée 50.51.52.53/32

Exemple :

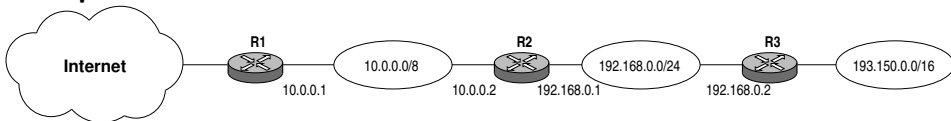


Table de R2 :

Destination/Préfixe	Routeur
10.0.0.0/8	0.0.0.0
192.168.0.0/24	0.0.0.0
193.150.0.0/16	192.168.0.2
0.0.0.0/0	10.0.0.1

CIDR et l'agrégation d'adresses (1)

Soient plusieurs entreprises qui, comme la précédente, se sont vues affecter de nombreuses adresses de classe C :

- 193.148.0.0/16 pour entreprise X
- 193.149.0.0/16 pour entreprise Y
- 193.150.0.0/16 pour l'entreprise précédente (E)
- 193.151.0.0/16 pour entreprise Z

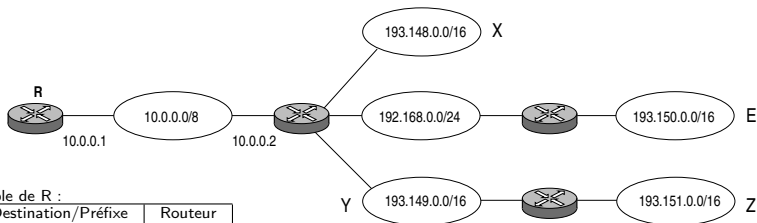


Table de R :

Destination/Préfixe	Routeur
10.0.0.0/8	0.0.0.0
192.168.0.0/24	10.0.0.2
193.148.0.0/16	10.0.0.2
193.149.0.0/16	10.0.0.2
193.150.0.0/16	10.0.0.2
193.151.0.0/16	10.0.0.2

CIDR et l'agrégation d'adresses (2)

Or,

- 193.148.0.0 = **11000001 100101**00 00000000 00000000
- 193.149.0.0 = **11000001 100101**01 00000000 00000000
- 193.150.0.0 = **11000001 100101**10 00000000 00000000
- 193.151.0.0 = **11000001 100101**11 00000000 00000000

Par **agrégation**, on regroupe les 4 entrées en une seule.

Nouvelle table de R :

Destination/Préfixe	Routeur
10.0.0.0/8	0.0.0.0
192.168.0.0/24	10.0.0.2
193.148.0.0/ 14	10.0.0.2

CIDR et agrégation d'adresse (3)

Il est possible d'agréger des blocs (incomplets) tout en ayant des routes différentes pour des sous-blocs :

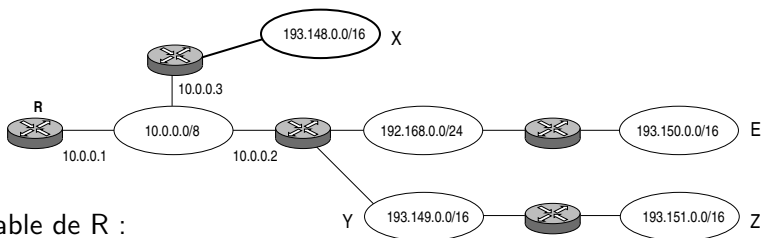


Table de R :

Destination/Préfixe	Routeur
10.0.0.0/8	0.0.0.0
192.168.0.0/24	10.0.0.2
193.148.0.0/14	10.0.0.2
193.148.0.0/16	10.0.0.3

pour réseaux E, Y et Z
pour réseau X

la dernière route est une exception à la route 193.148.0.0/14 !

L'adressage hors classe a été conçu pour être généralisé car :

- il réduit les tables de routage
- il est devenu difficile pour l'IANA (*Internet Assigned Numbers Authority*) d'allouer elle-même toutes les adresses IP

Blocs CIDR et délégation d'allocation

- l'ICANN (et avant lui l'IANA) a découpé les adresses disponibles en **blocs d'adresses continues**, partageant toutes un préfixe binaire
- puis a délégué la gestion/allocation de certains blocs aux 5 Registres Internet Régionaux (RIR) :



- AfriNIC : Afrique
 - APNIC : Asie et Pacifique
 - ARIN : Amérique du Nord
 - LACNIC : Amérique Latine et quelques îles des Caraïbes
 - RIPE NCC : Europe, Moyen-Orient et Asie Centrale
- Exemples :
 - 82.0.0.0/8 : RIPE NCC
 - 63.0.0.0/8 : ARIN
 - 154.0.0.0/8 : AfriNIC

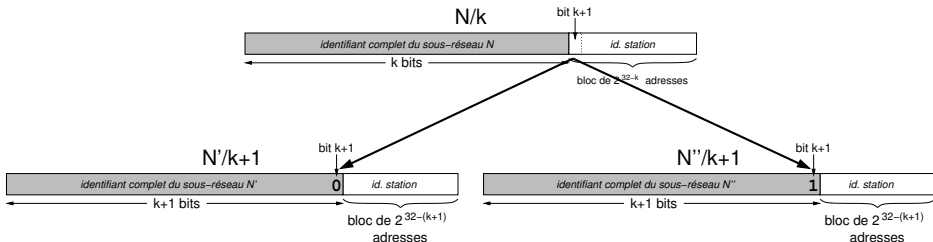
Voir <http://www.iana.org/assignments/ipv4-address-space> et RFC 1466

Blocs CIDR et délégation d'allocation (suite)

- les RIR attribuent ensuite des (parties de) blocs CIDR à des *National Internet Registry* (NIR), des *Local Internet Registry* (LIR), et à des FAI
- Exemple : 82.224.0.0/11 pour une partie du parc freebox de Free
- Ces registres/FAI gèrent eux-même leurs blocs d'adresses et attribuent des sous-blocs à leurs clients en fonction de leurs besoins

Partitionnement d'un bloc d'adresses : arbre binaire

Pour partitionner un bloc d'adresses N/k en sous-blocs :



- on le remplace par 2 sous-blocs disjoints $N'/k + 1$ et $N''/k + 1$, qui se distinguent par le $k + 1^{\text{e}}$ bit (à 0 dans un sous-bloc, et à 1 dans l'autre) :
- chaque sous-bloc peut ensuite être partitionné indépendamment
- ce qui forme un **arbre binaire** dont les feuilles (seulement) peuvent être affectées à des réseaux

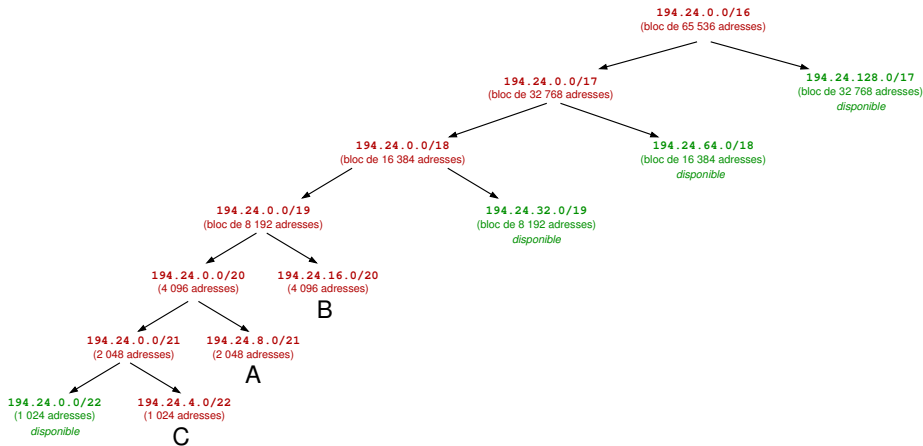
Même technique que le sous-adressage variable.

Exemple d'allocation de blocs par un FAI

Soit un FAI qui gère 194.24.0.0/16 :

- ① une entreprise A veut 2 000 adresses : obtient le bloc de 2 048 adresses 194.24.8.0/21
allant de 194.24.8.0 à 194.24.15.255
- ② puis une entreprise B veut 4 000 adresses : obtient le bloc de 4 096 adresses 194.24.16.0/20
allant de 194.24.16.0 à 194.24.31.255
- ③ puis une entreprise C veut 1 000 adresses : obtient le bloc de 1 024 adresses 194.24.4.0/22
allant de 194.24.4.0 à 194.24.7.255

Exemple d'allocation de blocs (arbre binaire)



- Alors que la croissance d'Internet a été exponentielle, le CIDR et le NAT ont permis de contourner les limites d'IPv4
- Mais en février 2011, un événement majeur a eu lieu : **le dernier bloc CIDR a été attribué par l'ICANN**
- Il ne reste que quelques adresses IPv4 encore *en stock* chez les registres

Il ne devrait plus se passer beaucoup de temps avant qu'IPv6 ne remplace définitivement IPv4.

Mais depuis le temps qu'on nous l'annonce. . .