

# Le routage dynamique

# Le routage dynamique

Routage statique

## En statique ?

- ▶ La configuration d'une route par défaut se fait de la façon suivante
  - ▶ Sur routeur Cisco : `Routeur(config)# ip route 0.0.0.0 0.0.0.0 163.13.100.1`
  - ▶ Sur Linux : `route add default gw 163.13.100.1`

Default gateway : 163.13.100.1  
168.13.0.0/24 subnetted  
168.13.1.0 via 168.13.100.1  
168.13.3.0 Ethernet 0  
168.13.2.0 via 168.13.100.2  
168.13.100.0 Ethernet 1

- ▶ Si un routeur associe d'abord une adresse de réseau de classes A, B ou C sur lequel elle réside avant de rechercher le sous-réseau spécifique, il est dit fonctionner **avec classe**. Sinon, il est dit fonctionner sans classe, ou **classless**
- ▶ Avec l'exemple précédent, avec classe la commande ping 168.13.200.1 échoue, sans class elle fonctionne (si une machine avec cette adresse existe )
  - ▶ Avec classe, il existe une entrée pour 168.13.0.0 de classe B mais je ne trouve pas de sous-réseau spécifique. Je n'utilise la route par défaut que s'il n'y a pas de correspondance

# Le routage dynamique

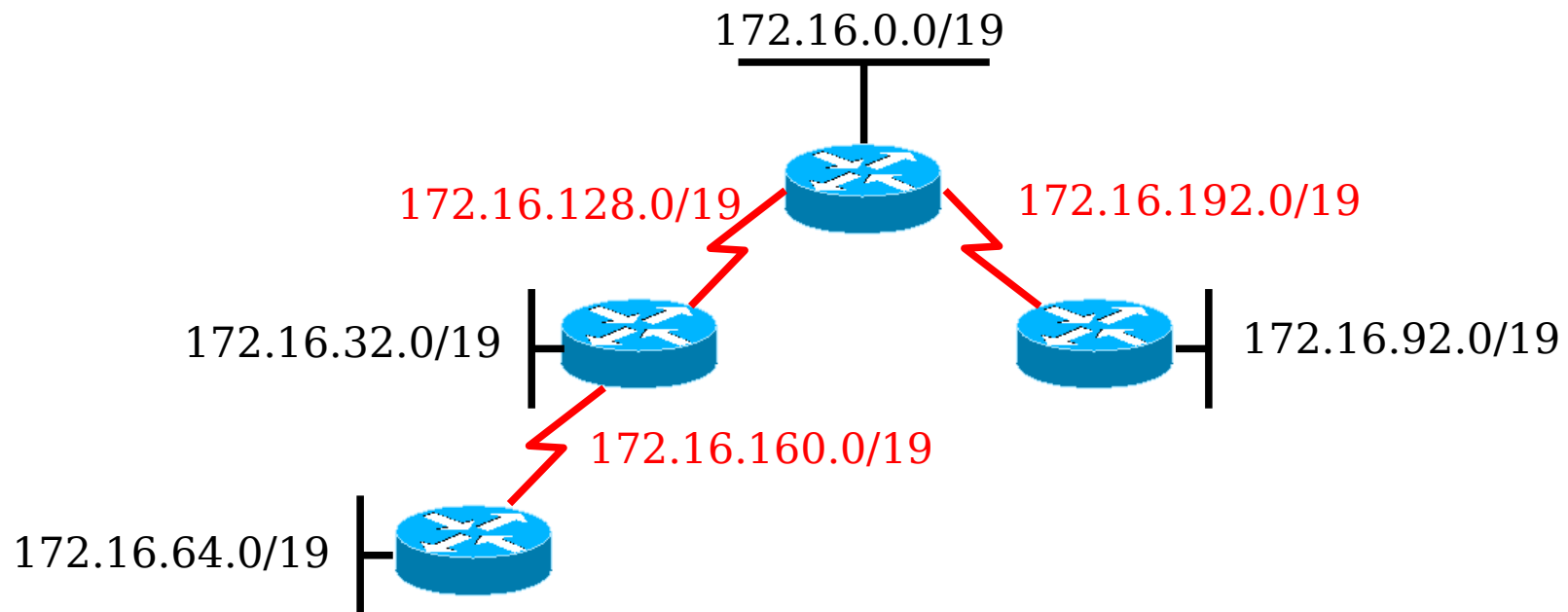
VLSM ?

# Présentation

- ▶ Lors de la fabrication de sous-réseaux, le nombre maximum de machines par sous-réseau est imposé par le découpage
  - ▶ Exemple : Avec le réseau 192.168.10.0/26, nous disposons de 4 sous-réseaux contenant chacun au plus 62 adresses disponibles
    - ▶ Attention : 4 sous-réseaux si on décide d'utiliser le sous-réseau 0. Sur un routeur Cisco, depuis la version 12.0 d'IOS, le subnet zero est utilisé par défaut
    - ▶ Si vous précisez un `no ip subnet-zero`, nous avons que 3 sous-réseaux de disponible
- ▶ Si, dans le réseau 192.168.10.0/24, nous sommes dans l'obligation de faire des sous-réseaux dont certains n'ont besoin que de 2 machines comme des liaisons point à point (PPP ou HDLC)
  - ▶ Gaspillage d'adresses IP !
- ▶ Solution : le VLSM (Variable Length Subnet Mask) permettant de faire des masques de sous-réseaux de longueur variable
  - ▶ Ainsi, il est possible d'ajuster au mieux la plage d'adresses IP disponible par sous-réseau

## Exemple d'utilisation du VLSM (1/3)

- Soit le schéma suivant : besoin de 7 sous-réseaux



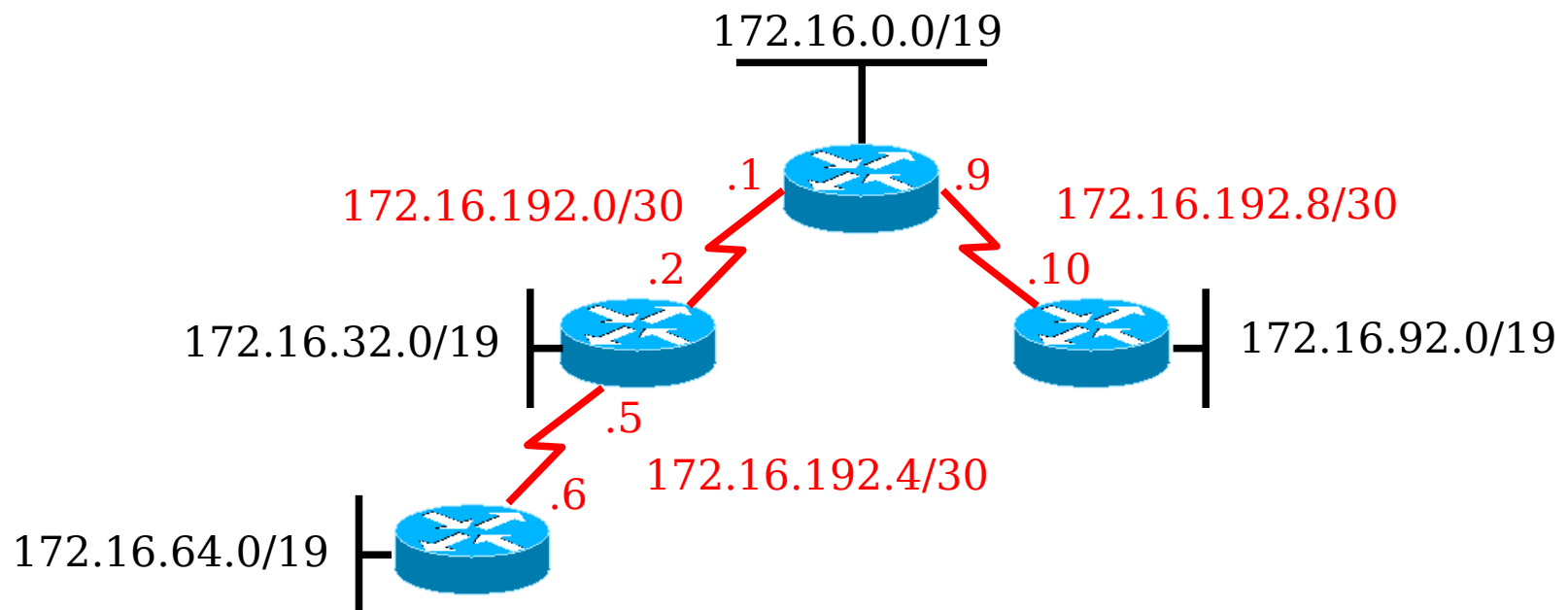
Dans chaque sous-réseau, nous disposons au plus de 16382 IP  
dont 3\*16380 IP perdues pour les liaisons séries !

## Exemple d'utilisation du VLSM (2/3)

sous-réseau 0	172.16.0.0	/19
sous-réseau 1	172.16.32.0	/19
sous-réseau 2	172.16.64.0	/19
sous-réseau 3	172.16.92.0	/19
sous-réseau 4	172.16.128.0	/19
sous-réseau 5	172.16.160.0	/19
sous-réseau 6	172.16.192.0	/19
sous-réseau 7	172.16.226.0	/19
sous sous-réseau 0	172.16.192.0	/30
sous sous-réseau 1	172.16.192.4	/30
sous sous-réseau 2	172.16.192.8	/30

## Exemple d'utilisation du VLSM (3/3)

- Soit le schéma suivant : besoin de 7 sous-réseaux





## Et le CIDR ?

- ▶ Classless InterDomain Routing (CIDR) est une convention définie dans le RFC 1817
  - ▶ Demande l'agrégation de plusieurs numéros de réseaux en une seule entité de routage
- ▶ Sans le CIDR, un provider qui gère les réseaux 198.1.1.0 à 198.1.255.0 doit définir une route vers chacun de ces réseaux de classe C, soit 255 routes
- ▶ Avec le CIDR, il fera une route unique pour le 198.1.0.0/16

# Agrégation des routes

- ▶ Utilisation du VLSM augmente le nombre de sous-réseau utilisé et donc les tables de routages
- ▶ Afin de limiter cette augmentation, les routeurs peuvent avoir recours à l'agrégation de routes



```

11000000.10101000.01100010.00000
11000000.10101000.01100011.00000
11000000.10101000.01100101.00000
11000000.10101000.01101001.00000
  
```

# Comment utiliser les agrégations ?

- ▶ Choisir des adresses de sous-réseaux proches sur un même routeur afin de pouvoir faire de l'agrégation
- ▶ Lors du choix du VLSM, utiliser un sous-réseau libre pour faire un nouveau découpage en sous sous-réseau

# Le routage dynamique

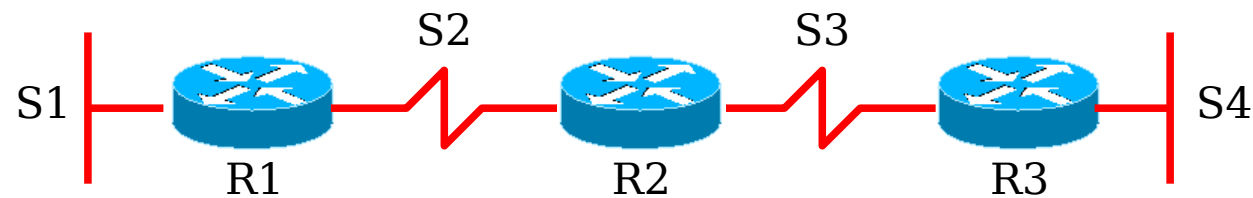
Algorithmes de routage à vecteur distance

# Le routage

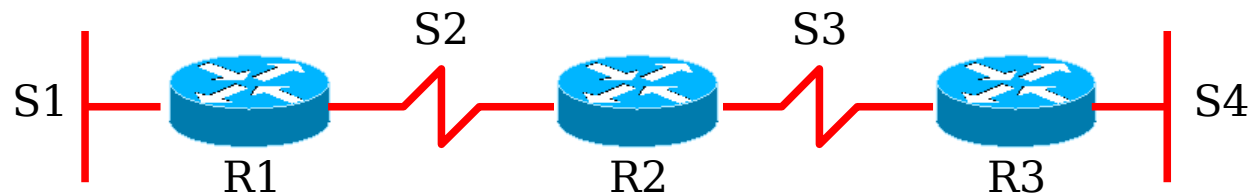
- ▶ Internet est un ensemble de systèmes autonomes (SA) interconnectés par des routeurs
- ▶ Chaque SA possède son propre algorithme de routage, dit IGP (Interior Gateway Protocol)
- ▶ Entre chaque SA, les protocoles d'échanges d'information de routage sont dit EGP (Exterior Gateway Protocol)

# Fonctionnement

- ▶ Les mises à jour prennent un certain temps avant d'atteindre tous les routeurs
- ▶ Pendant ce temps, des routes peuvent disparaître et d'autres apparaître
- ▶ Ce temps est dit être le temps de **convergence**
- ▶ Quelque soit l'algorithme utilisé, chacun est sujet au phénomène du **comptage à l'infini**
  - ▶ Considérons 3 routeurs



# Comptage à l'infini



► Table de routage de R1 :

S1 et S2 : distance 0  
S3 : distance 1  
S4 : distance 2

► Table de routage de R3 :

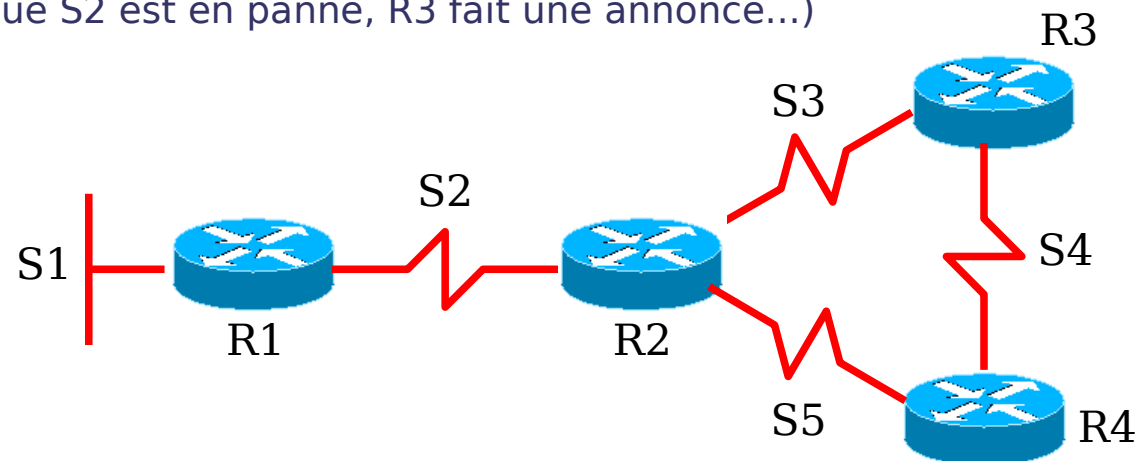
S3 et S4 : distance 0  
S1 : distance 2  
S2 : distance 1

► Après une coupure de S2 : R1 fait sa mise à jour donc n'envoie plus de route pour S1 vers R2

- Or R3 connaît une route pour S1 donc R2 va la prendre en compte
- Pour R2 : distance vers S1 sera 3, puis 4, puis ...
- Pour R3 : distance vers S1 sera 4, puis 5, puis ...

# Split horizon

- ▶ Une règle qui interdit à tout routeur d'annoncer un préfixe réseau via l'interface par laquelle il a appris l'existence de celui-ci ou par laquelle il passe pour accéder à ce réseau
- ▶ Pourquoi ? Supposons que S2 ne soit plus disponible, R2 met à jour sa propre table et l'envoie à R3. En même temps, R3 envoie sa table à R2
  - ▶ R2 apprend qu'il existe une route pour S2 par R3
  - ▶ R3 apprend que S2 n'existe plus
- ▶ Avec le split horizon, dans ce cas, nous n'aurions pas le comptage à l'infini
- ▶ Mais le split horizon n'empêche pas toujours le comptage à l'infini comme dans l'exemple suivant (suppose que S2 est en panne, R3 fait une annonce...)





# Split horizon avec Poison-reverse

- ▶ Cisco utilise une variante du Split horizon.
- ▶ Quand le réseau est stable, utilisation du split horizon classique
- ▶ Quand une route vers un réseau est annoncée avec une distance infinie, le destinataire de la mise à jour l'annonce aussi avec une métrique infinie sur toutes ces interfaces, y compris celles protégées par le Split-horizon

# Temporisateur de retenue Hold-down

- ▶ Le comptage à l'infini peut être résolu par cette fonction
- ▶ Lorsqu'un routeur prend connaissance de l'indisponibilité d'une route
  - ▶ Il doit ignorer toute information concernant un chemin vers ce sous-réseaux pendant un durée égale au temporisateur Hold-down
- ▶ Ainsi, si une annonce arrive sur un routeur qui vient d'annoncer une défaillance pour un même sous-réseau
  - ▶ Ce routeur va atteindre avant de prendre en compte cette annonce
  - ▶ Ainsi, l'annonce de distance infinie aura le temps de se propager

# Fonction Route-poisoning

- ▶ Dès qu'une ligne tombe en panne, un protocole de routage peut choisir de simplement arrêter de l'annoncer
  - ▶ Mais comment distinguer une panne réelle d'une absence de mise à jour perdu à cause de bruit sur la ligne ?
- ▶ La fonction Route-poisoning impose au routeur de faire une annonce pour la ligne en panne avec une métrique infinie sur les mêmes interfaces que celles par lesquelles elle a pu être annoncée

## Mises à jour déclenchées

- ▶ Dès qu'un routeur remarque un changement d'état sur le sous-réseau directement connecté, il expédie immédiatement ses nouvelles informations.
- ▶ Cette technique couplée au routing-poisoning et au Poison-reverse accélère la convergence des algorithmes de routages

# Expiration et redondance de routes ?

- ▶ Une absence d'annonce de la part d'un routeur voisin qui se prolongerait sur un nombre prédéfini d'intervalles de mises à jour entraîne la suppression des routes qu'ils avaient annoncées
- ▶ Si plusieurs routes sont disponibles pour un même sous-réseau, celle avec la meilleur métrique sera choisie
- ▶ Si plusieurs routes de métriques équivalentes pour un même sous-réseau sont découvertes, le routeur conserve celle déjà enregistrée
  - ▶ Il est possible d'ajouter plusieurs routes de même coût dans la même table et de répartir la charge à travers ces lignes au lieu de n'utiliser qu'une seule voie

# Les algorithmes existants

- ▶ Les principaux algorithmes de routage à vecteur distance sont
  - ▶ RIP version 1
  - ▶ RIP version 2
  - ▶ IGRP
- ▶ Leurs caractéristiques

<b>Fonctionnalités</b>	<b>RIP</b>
Temporisateur de mise à jour	30 secondes
Métrique	compteur de sauts
Temporisateur Hold-down	180 secondes
Mises à jour déclenchées	Oui
Masque de sous-réseau envoyé	Non pour RIP v1, Oui pour RIP v2

# Le routage dynamique

RIP v2

# Historique : RIP v1

- ▶ RIP v1 est un algorithme de routage, considéré comme un IGP, basé sur des vecteurs distances. Chaque table de routage est complètement diffusée sur le réseau à intervalle de temps pré-déterminé (par défaut 30s). La distance maximale géré par cet algorithme est de 15 hops (16 = infini).
- ▶ Quand un routeur reçoit des informations relatives à un réseau sur une interface appartenant au même réseau, mais à un sous-réseau différent, le routeur applique le masque de sous-réseau de cette interface



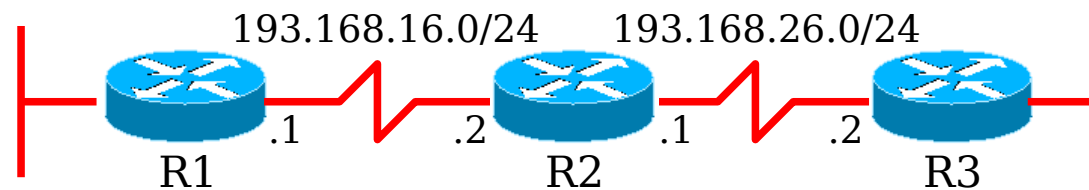
## Limites de RIP v1

- ▶ N'envoie pas les masques de sous-réseau dans ses updates
- ▶ Les updates sont broadcastés
- ▶ Pas d'authentification
- ▶ Ne supporte ni le VLSM ni le CIDR (Classless InterDomain Routing)
- ▶ Utilise Split Horizon et le holddown pour détecter les boucles

## RIP v2

- ▶ Supporte le VLSM car transmission d'un masque de sous-réseau avec les routes
- ▶ Authentification de la source d'une mise à jour de routage peut se fonder sur un texte clair ou sur un texte crypté avec l'algorithme MD5
- ▶ Mise à jour de routage par adresse multicast contrairement à RIP v1 qui diffuse ses mises à jour via l'adresse 255.255.255.255
  - ▶ RIP v2 transmet à l'adresse IP de destination multicast 224.0.0.9, réservé pour une utilisation par RIP v2

# Configuration de RIP

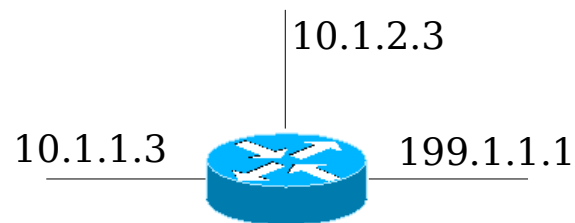


- ▶ Pour activer le protocole de routage RIP sur un routeur, il suffit d'utiliser la commande `router rip` en mode de configuration global, puis de déclarer les réseaux connectés directement avec la commande `network`

```
Sur le routeur R2
Router(config)# router rip
Router(config-router)# network 193.168.16.0
Router(config-router)# network 193.168.26.0
```

- ▶ Chaque commande `network` active RIP ou IGRP sur un ensemble d'interfaces
  - ▶ envoi périodique de mises à jour
  - ▶ écoute régulière par le routeur des mises à jour reçues par cette interface
  - ▶ inclusion du masque de sous-réseau accessible via cette interface

## Autre exemple



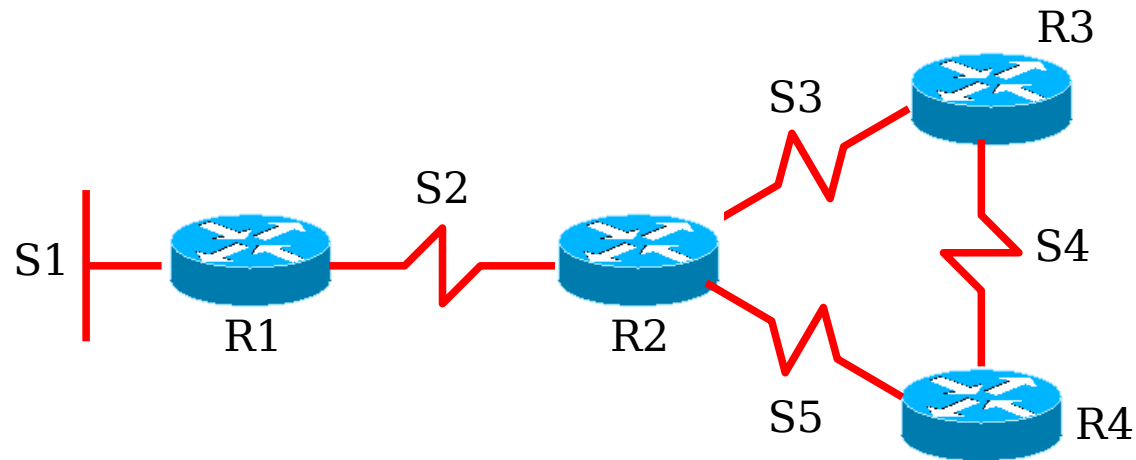
```
router rip
network 10.0.0.0
network 199.1.1.0
```

- ▶ network 10.0.0.0 affecte les 2 interfaces concernées
- ▶ La commande network attend en paramètre un numéro de réseau et non pas un sous-réseau.
  - ▶ Dans le cas où vous tapez un numéro de sous-réseau, il sera automatiquement transformé en numéro de réseau
- ▶ Attention : si le routeur fonctionne en no ip classless, c'est à dire que le routage se fait suivant la classe, un paquet à destination de 10.1.3.3 ne sera pas routé sur la route par défaut car aucun réseau 10.1.3.0/24 n'est connecté au routeur
  - ▶ En ip classless, il sera par contre routé sur la route par défaut

# Configuration des options de RIP (1/3)

- ▶ RIP utilise les techniques suivantes pour éviter les boucles et accélérer la convergence
  - ▶ Split horizon
  - ▶ Split horizon with poison reverse (ou Poison reverse)
  - ▶ Holddown
  - ▶ Comptage à l'infini (16 est l'infini)
  - ▶ Mise à jour déclenchée
- ▶ Pour désactiver le split horizon :
  - ▶ Router(config-if)# **no ip split-horizon**
- ▶ Par défaut la valeur du holddown est de 180 secondes. Dans certains cas, il peut être utile de diminuer cette valeur
  - ▶ en configuration router : **timers basic** *update invalid holddown flush*
  - ▶ Faire un **no timers basic** pour retrouver les valeurs par défaut

## Configuration des options de RIP (2/3)



- ▶ Dans l'exemple précédent, un holddown de 90 secondes est suffisant. Avec des mises à jour toutes les 30 secondes, si S2 disparaît, R2 va recevoir une nouvelle route pour S2 de la part de R3 (par exemple) mais si R2 attend 90 secondes avant de faire sa mise à jour, S2 aura disparu des tables de routage de R4 puis de R3

## Configuration des options de RIP (3/3)

- ▶ Par défaut les mises à jour sont faites toutes les 30 secondes
  - ▶ Valeur modifiable par : Router(config-router)# **update-timer** *seconds*
- ▶ Par la commande `network`, on « active » pour le routage toutes les interfaces sur le réseau spécifié
  - ▶ Pour qu'une interface n'envoie plus de mises à jour : Router(config-router)# **passive-interface** *interface* (**Attention** : elle continue à les recevoir)
- ▶ Comme RIP envoie ses mises à jour par broadcast, il se peut que le routeur soit connecté à un routeur voisin qui n'accepte pas les broadcast de part le protocole d'interconnexion qu'il utilise. Dans ce cas, on peut spécifier directement l'adresse de ce voisin pour lui envoyer en unicast les mises à jour
  - ▶ Router(config-router)# **neighbor** *ip\_address*
- ▶ Par défaut, les routeurs Cisco reçoivent les paquets RIP v1 et v2 mais n'envoie que des paquets RIP v1. Pour spécifier quel est le protocole à utiliser sur les interfaces
  - ▶ Router(config-router)# **version** {1|2}
  - ▶ Router(config-if)#**ip rip send version** 1 OU **ip rip send version** 2 OU **ip rip send version** 1 2
  - ▶ Router(config-if)#**ip rip receive version** 1 OU ...

# Équilibrage de charge dans RIP

- ▶ RIP est capable de gérer jusque 6 routes d'un même coût
- ▶ Quand un routeur reçoit une information via un algorithme de routage, il choisit toujours l'information avec la plus petite distance administrative (le coût)

Protocole	Distance
Directement connecté	0
Route statique	1
Enhanced IGRP summary route	5
External BGP	20
Internal Enhanced IGRP	90
IGRP	100
OSPF	110
Is-IS	115
RIP	120
EIGRP external route	170
Internal BGP	200



# Vérification des protocoles utilisés

- ▶ Pour visualiser les routes : Router# `show ip route`
- ▶ Pour visualiser le protocole de routage utilisé: Router# `show ip protocols`
- ▶ Pour visualiser le détail d'une interface:
  - ▶ Router# `show interface interface`
  - ▶ *ou* Router# `show interface` pour visualiser toutes les interfaces
- ▶ Pour visualiser un interface ip
  - ▶ Router# `show ip interface`
  - ▶ *ou* Router# `show ip interface brief` pour visualiser uniquement les informations importantes de toutes les interfaces
- ▶ Pour visualiser la base de données RIP du routeur
  - ▶ Router# `show ip rip database`