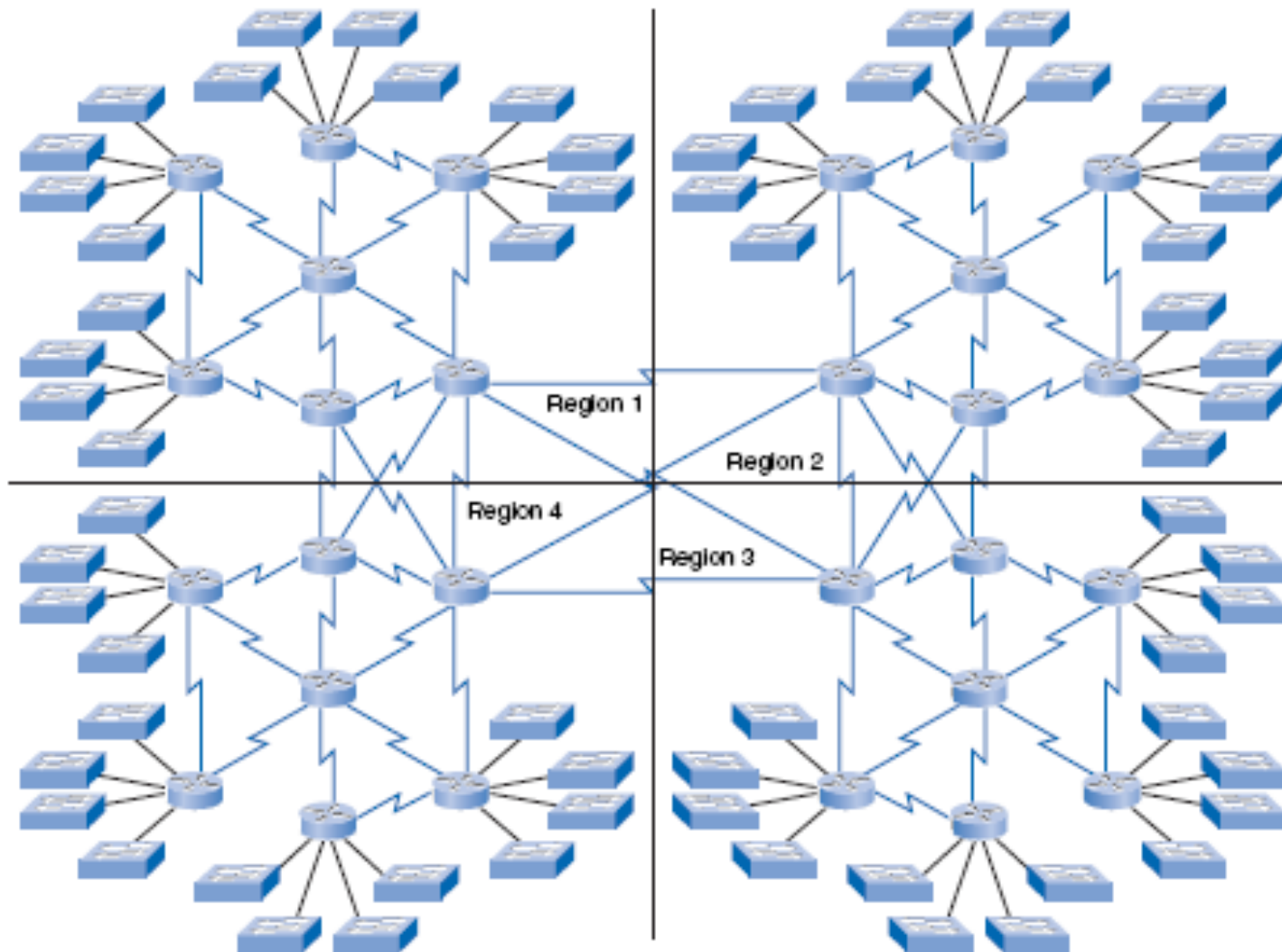


# Interconnexion à grande échelle

- Le routage dynamique
- Le protocole RIP version 1



Imaginez-vous gérer les liens statiques dans un réseau pareil

Qu'est-ce que se passe si un lien tombe à 3h du matin ?

# Vecteur de distances

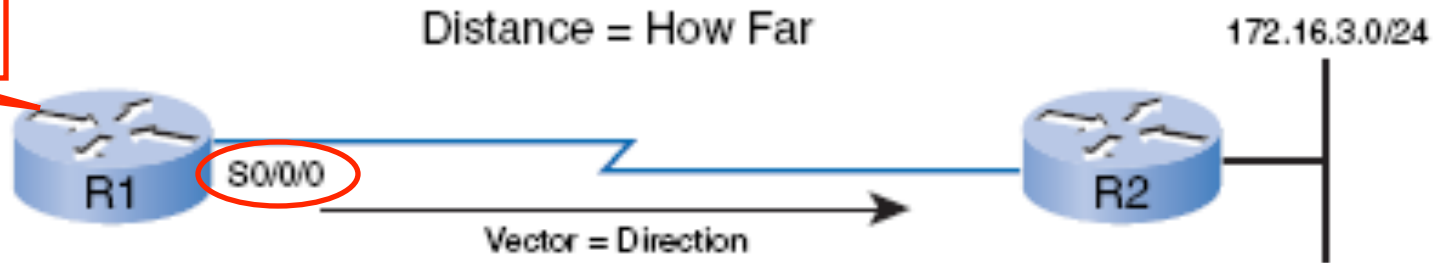
## ► **Vecteur de distances**

- Les routes sont annoncées comme **vecteurs de distance et direction**.
- La **distance** est définie selon une métrique
  - Ex : le nombre de sauts (hop count)
- La **direction** indique simplement :
  - L'adresse du prochain routeur ou
  - L'interface de sortie.
- Le protocole de routage
  - Ne connaît pas la topologie du réseau.
  - La seule information qu'il détient est l'information de routage reçue de ses voisins.



# Vecteur de distances

Je peux atteindre  
172.16.3.0/24 en  
un saut à partir de  
ma S0/0/0.



For R1, 172.16.3.0/24 is one hop away (distance).  
It can be reached through S0/0/0 (vector).

Quelle est la **Distance** jusqu'à 172.16.3.0/24 ?

1 saut (hop)

Quelle est la **Direction** ?

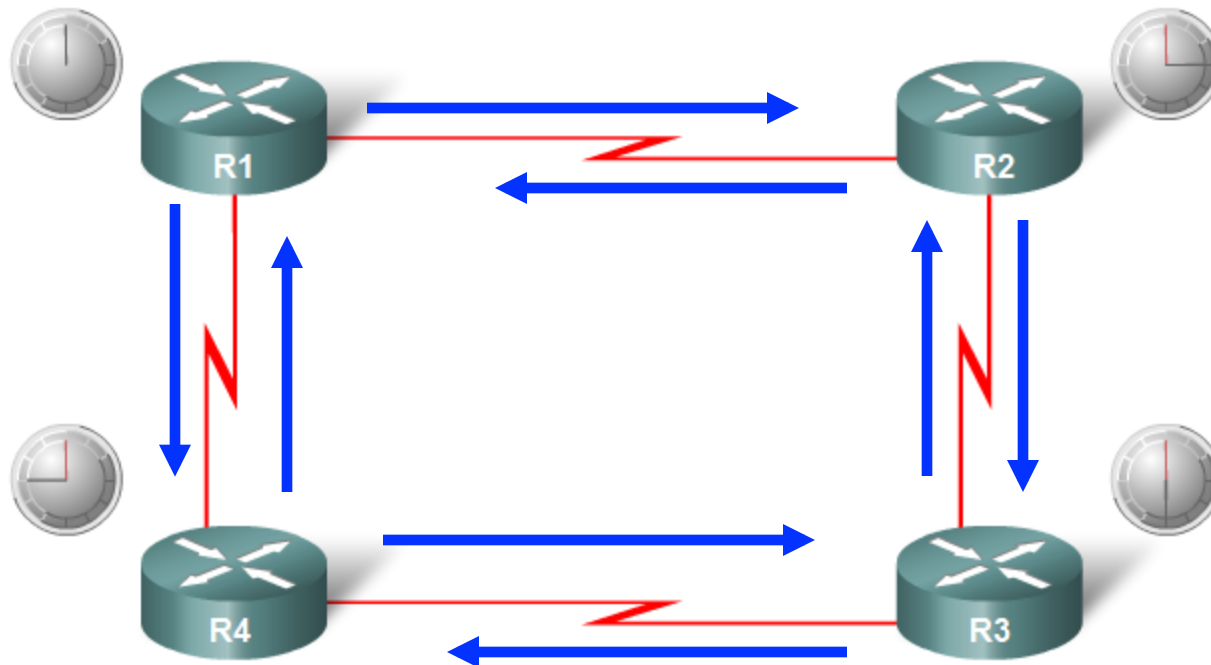
S0/0/0

R1 connaît la carte topologique du réseau ?

Non, seulement la distance et la direction !

# Opération des protocoles à vecteur de distances

Timer  
Expires



## Mises à jour régulières

Certains protocoles diffusent la totalité des tables de routage aux voisins (RIP and IGRP)

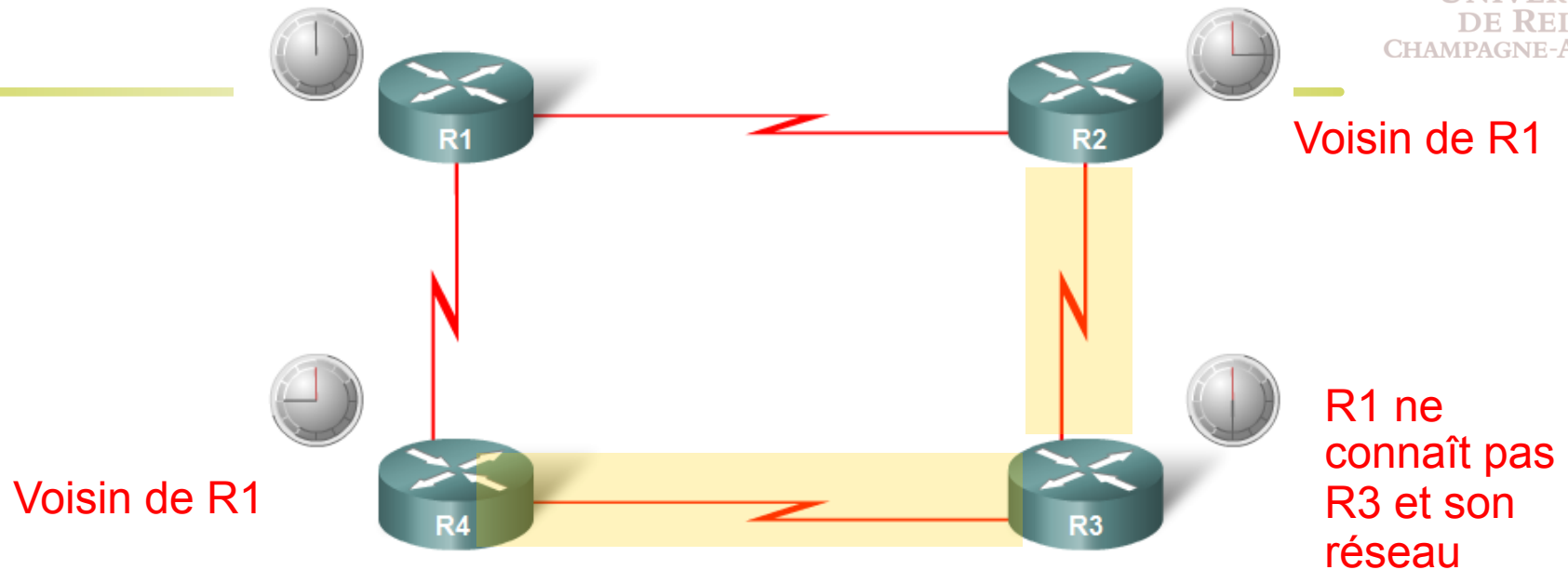
Intervalle de 30 seconds pour RIP

## Inefficace :

Gaspillage de bande passante et de CPU

Envoi régulier même si aucun changement a eu lieu

# Opération des protocoles à vecteur de distances



## Voisins ?

Les voisins sont les routeurs qui :

- Partagent un lien

- Utilisent le même protocole de routage

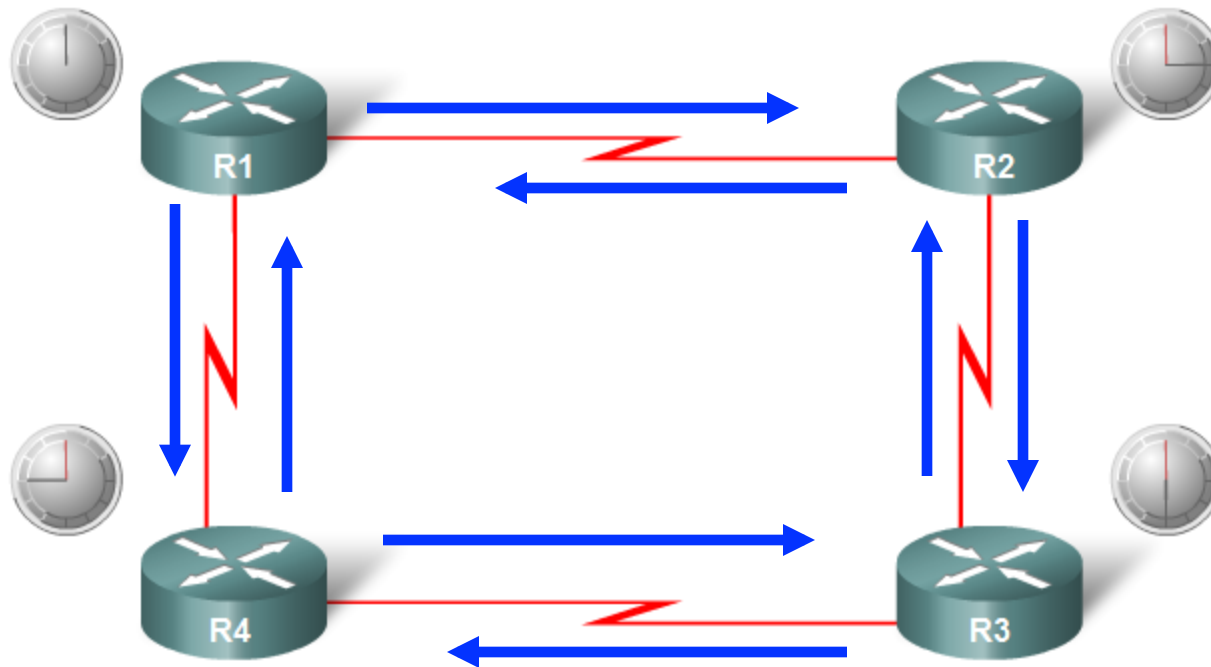
Quelles adresses un routeur connaît avant d'une mise à jour ?

- Les adresses de ses interfaces

- Les adresses de ses voisins

# Opération des protocoles à vecteur de distances

Timer  
Expires



**Les protocoles de routage utilisent**

Mises à jour par **Broadcast** (255.255.255.255)

Mises à jour par **Multicast**

**Les voisins doivent traiter les mises à jour**

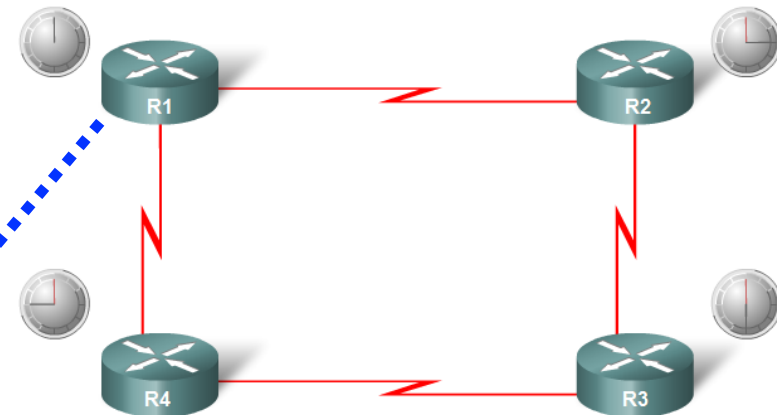
Si un voisin ne tourne pas un protocole de routage, il finira par jeter le message

# Algorithmes de Routage

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

Step 0 : Initialize  
 $d(s) := 0; d(v) := +\infty \forall v \in V \setminus \{s\}; \pi(v) = v \forall v \in V; Q := V; i := 1$   
 Step 1 : Select the node  
 If  $Q = \emptyset$ , then go to step 3, else select the node  $v$  from the head of  $Q$   
 Step 2 : Search the Path (let  $v$  be the initial point)  
 If  $d(u) > d(v) + l((v, u))$  for all path  $(v, u)$ , then  $d(u) = d(v) + l((v, u))$ ,  $\pi(u) = v$   
 $\rightarrow$  Step 1  
 Step 3: judgement  
 $i \leftarrow i + 1$   
 If  $i < n$ , then  $Q \leftarrow V$  and go to step 1,  
 else check whether triangle inequality\* is satisfied or not on all paths.  
 If any paths "A" not satisfied the triangle inequality, there is the negatively circuit including the path "A".

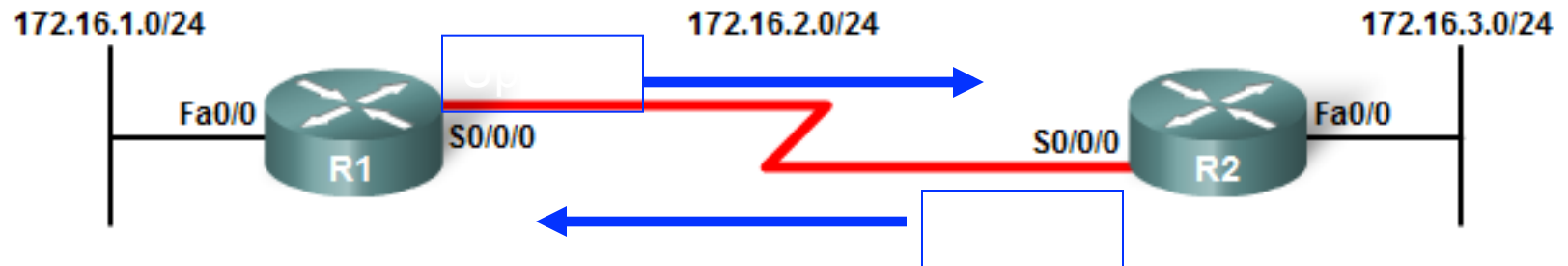
\* Triangle inequality  
 Let  $X$  be linear space,  
 $\|u+v\| \leq \|u\| + \|v\|$  for  $u, v \in X$



L'algorithme d'un protocole de routage est responsable par la **construction** et la **mise à jour** de la table de routage



# Algorithmes de routage

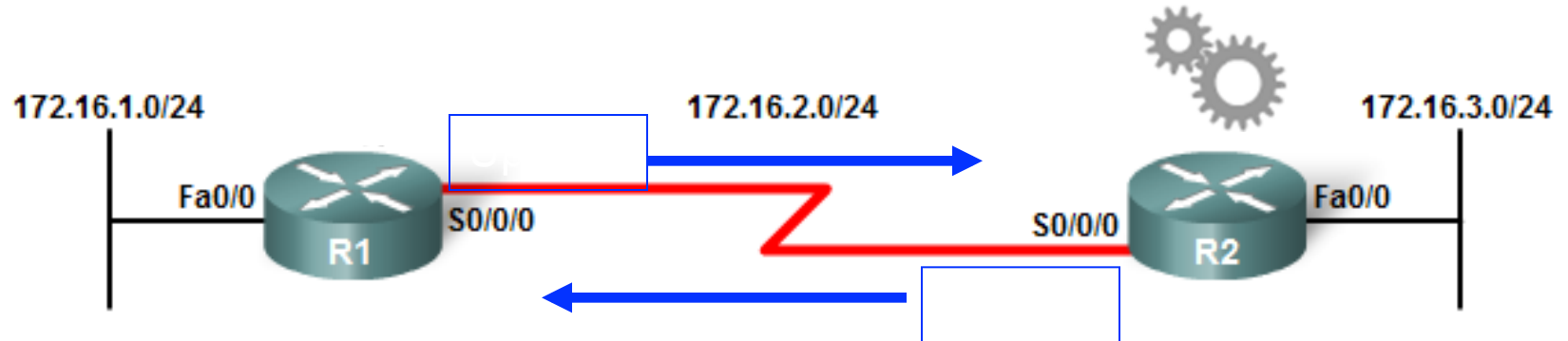


| Network       | Interface | Hop |
|---------------|-----------|-----|
| 172.16.1.0/24 | Fa0/0     | 0   |
| 172.16.2.0/24 | S0/0/0    | 0   |
|               |           |     |

| Network       | Interface | Hop |
|---------------|-----------|-----|
| 172.16.2.0/24 | S0/0/0    | 0   |
| 172.16.3.0/24 | Fa0/0     | 0   |
|               |           |     |

L'algorithme envoie et reçoit des mises à jour

# Algorithmes de Routage



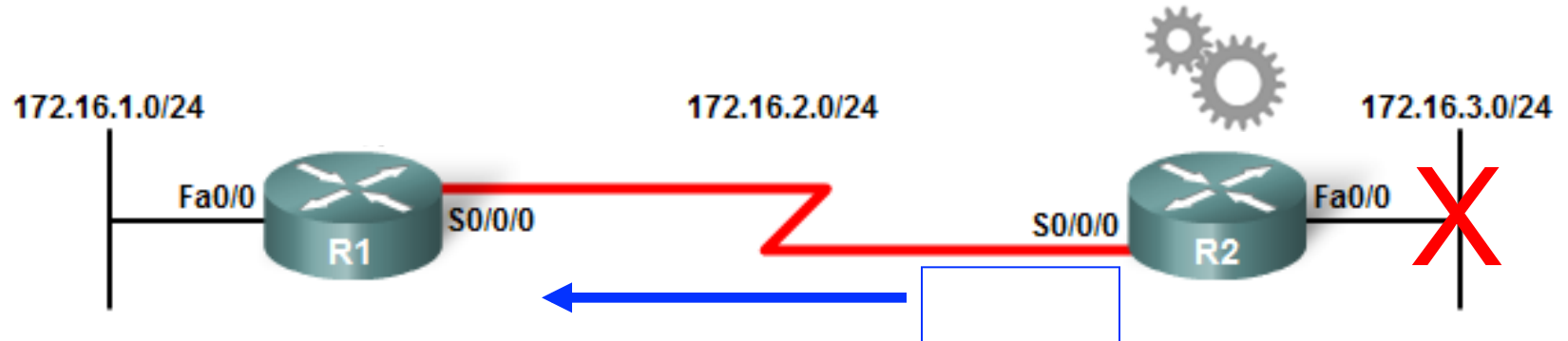
| Network       | Interface | Hop |
|---------------|-----------|-----|
| 172.16.1.0/24 | Fa0/0     | 0   |
| 172.16.2.0/24 | S0/0/0    | 0   |

| Network       | Interface | Hop |
|---------------|-----------|-----|
| 172.16.2.0/24 | S0/0/0    | 0   |
| 172.16.3.0/24 | Fa0/0     | 0   |

Dans chaque routeur, l'algorithme :

travaille indépendamment afin de mettre à jour la table de routage

# Algorithmes de Routage



| Network                  | Interface         | Hop          |
|--------------------------|-------------------|--------------|
| 172.16.1.0/24            | Fa0/0             | 0            |
| 172.16.2.0/24            | S0/0/0            | 0            |
| <del>172.16.3.0/24</del> | <del>S0/0/0</del> | <del>1</del> |

| Network                  | Interface        | Hop          |
|--------------------------|------------------|--------------|
| 172.16.2.0/24            | S0/0/0           | 0            |
| <del>172.16.3.0/24</del> | <del>Fa0/0</del> | <del>0</del> |
| 172.16.1.0/24            | S0/0/0           | 1            |

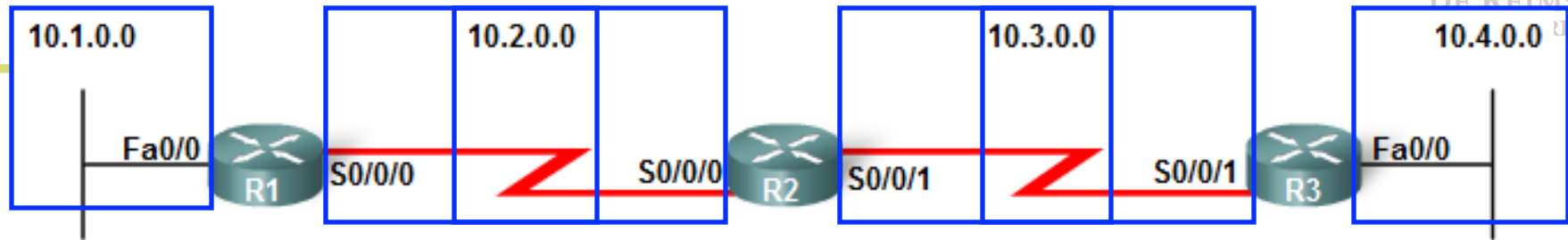
Dans chaque routeur, l'algorithme :

Détecte et réagit aux changements de topologie

# Découverte d'un Réseau

- Cold Start
- Échange initiale
- Échange d'informations de routage

# Cold Start



| Network | Interface | Hop |
|---------|-----------|-----|
|         |           |     |
|         |           |     |
|         |           |     |
|         |           |     |

| Network | Interface | Hop |
|---------|-----------|-----|
|         |           |     |
|         |           |     |
|         |           |     |
|         |           |     |

| Network | Interface | Hop |
|---------|-----------|-----|
|         |           |     |
|         |           |     |
|         |           |     |
|         |           |     |

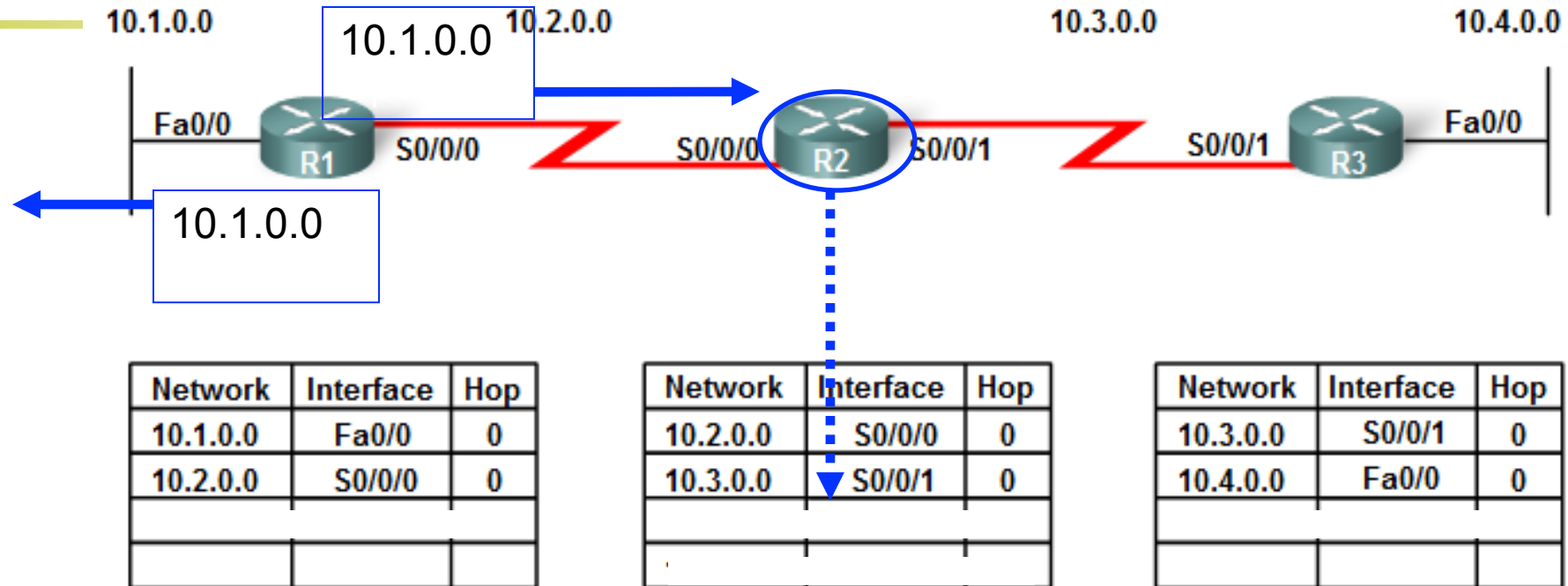
**La découverte du réseau** est la première chose qu'un protocole de routage doit faire

**Note :** La totalité des tables de routage est envoyée

Tout d'abord : On ne connaît que les réseaux directement connectés



# Échange Initial



**R1:** (envoie tout sa table de routage sauf 10.2.0.0 à cause du split horizon)

**Envoie** une mise à jour sur 10.1.0.0 via Serial 0/0/0 avec une métrique 1

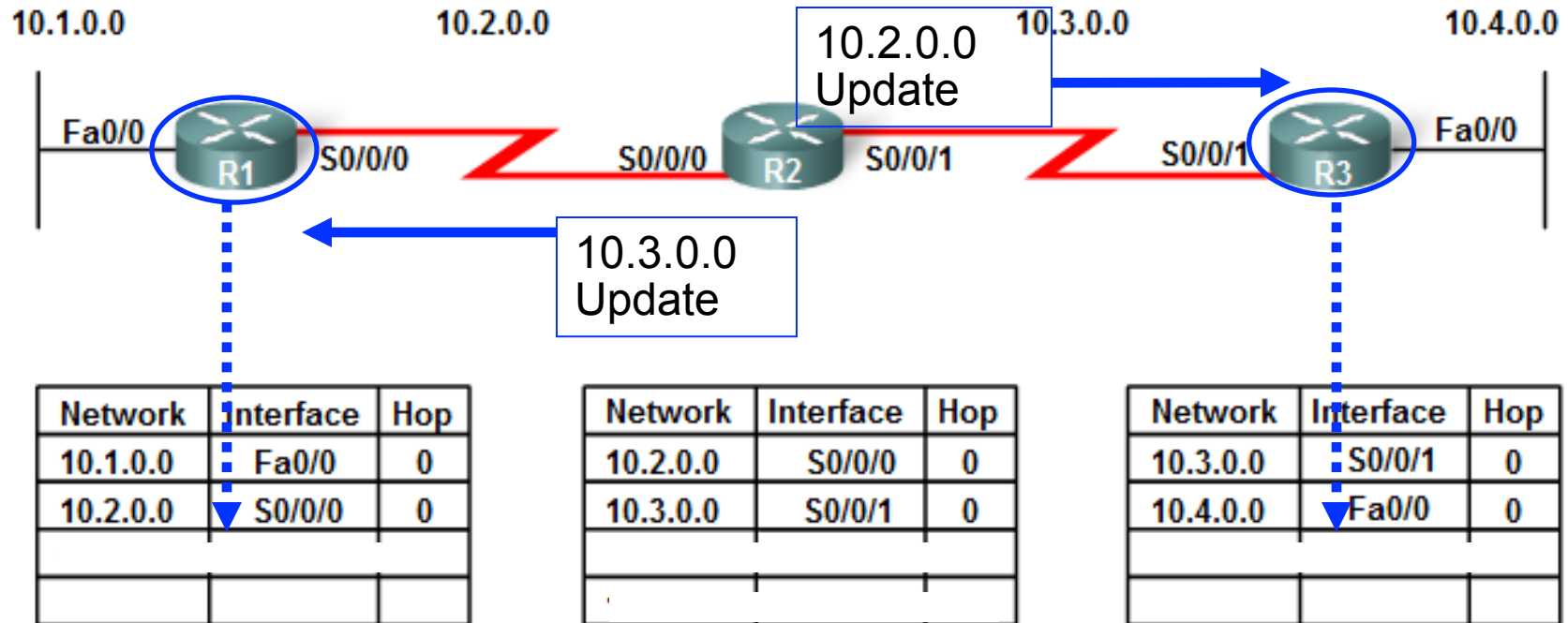
**Envoie** une mise à jour sur 10.2.0.0 via FastEthernet 0/0 avec une métrique 1

**R2**

**Reçoit** la mise à jour de R1 sur 10.1.0.0 sur son Serial 0/0/0 avec une métrique 1

**Enregistre** le réseau 10.1.0.0 dans la table de routage avec une métrique 1

# Échange Initiale



**R2** (au même temps que R1):

**Envoie** une mise à jour sur 10.3.0.0 via Serial 0/0/0 avec une métrique 1

**Envoie** une mise à jour sur 10.2.0.0 via Serial 0/0/1 avec une métrique 1

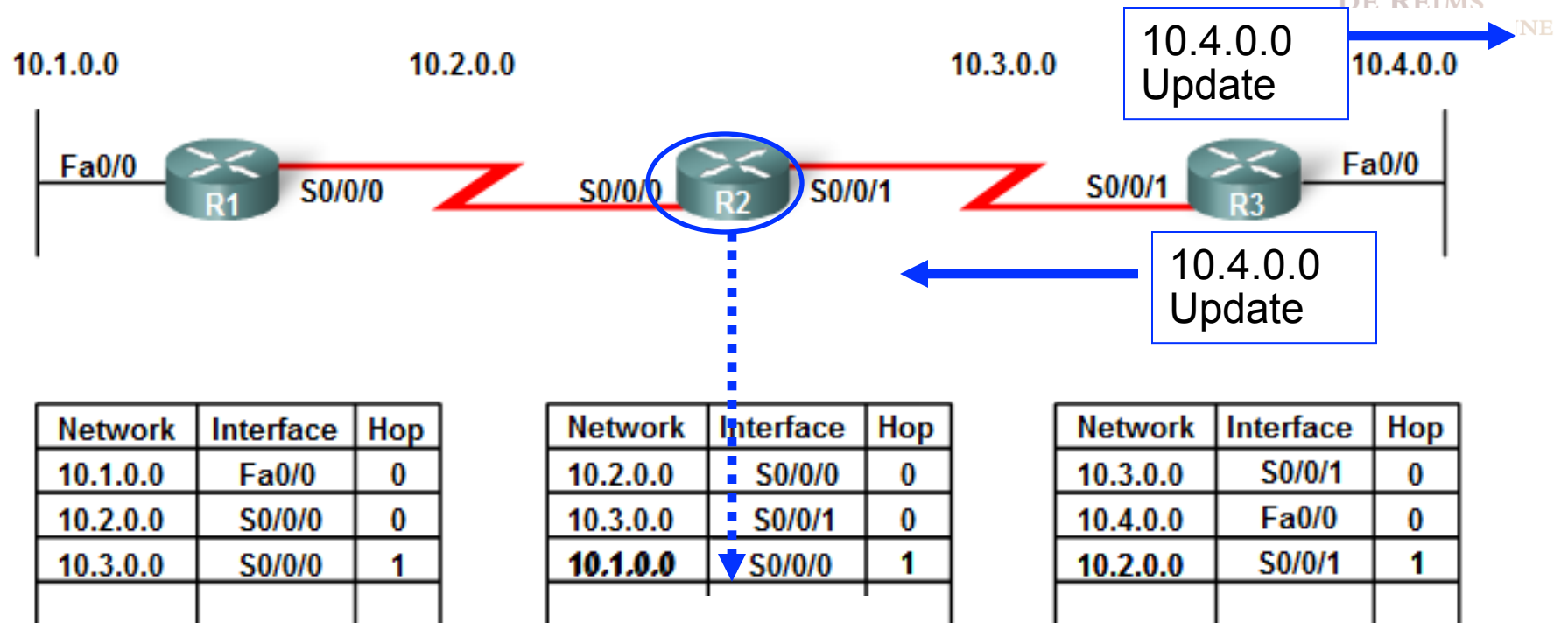
**R1** **Reçoit** la mise à jour de R2 sur 10.3.0.0 sur son Serial 0/0/0 avec une métrique 1

**Enregistre** le réseau 10.3.0.0 dans la table de routage avec une métrique 1

**R3** **Reçoit** la mise à jour de R2 sur 10.2.0.0 sur son Serial 0/0/1 avec une métrique 1

**Enregistre** le réseau 10.2.0.0 dans la table de routage avec une métrique 1

# Échange Initiale



**R3:** (Au même temps que R1 et R2)

**Envoie** une mise à jour sur 10.4.0.0 via Serial 0/0/1 avec une métrique 1

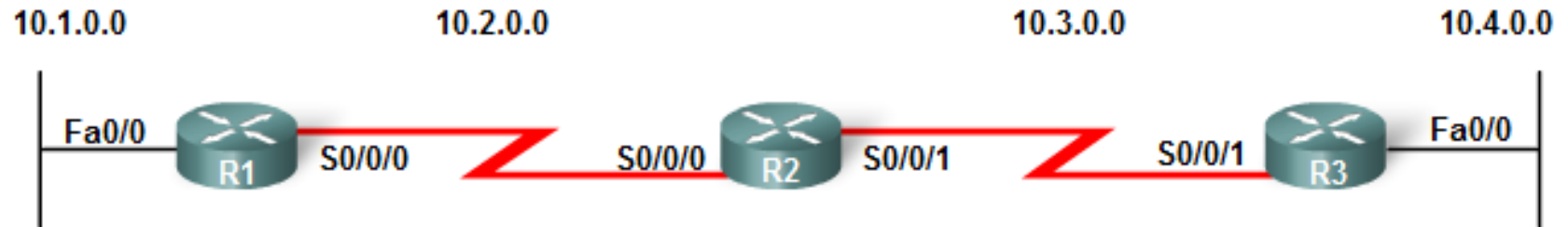
**Envoie** une mise à jour sur 10.3.0.0 via FastEthernet 0/0 avec une métrique 1

**R2 Reçoit** la mise à jour de R3 sur 10.4.0.0 sur son Serial 0/0/1 avec une métrique 1

**Enregistre** le réseau 10.4.0.0 dans la table de routage avec une métrique 1



# Échange Initial



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
|          |           |     |

| Network         | Interface | Hop |
|-----------------|-----------|-----|
| 10.2.0.0        | S0/0/0    | 0   |
| 10.3.0.0        | S0/0/1    | 0   |
| <b>10.1.0.0</b> | S0/0/0    | 1   |
| 10.4.0.0        | S0/0/1    | 1   |

| Network         | Interface | Hop |
|-----------------|-----------|-----|
| 10.3.0.0        | S0/0/1    | 0   |
| 10.4.0.0        | Fa0/0     | 0   |
| <b>10.2.0.0</b> | S0/0/1    | 1   |
|                 |           |     |

Le réseau a-t-il convergé ?

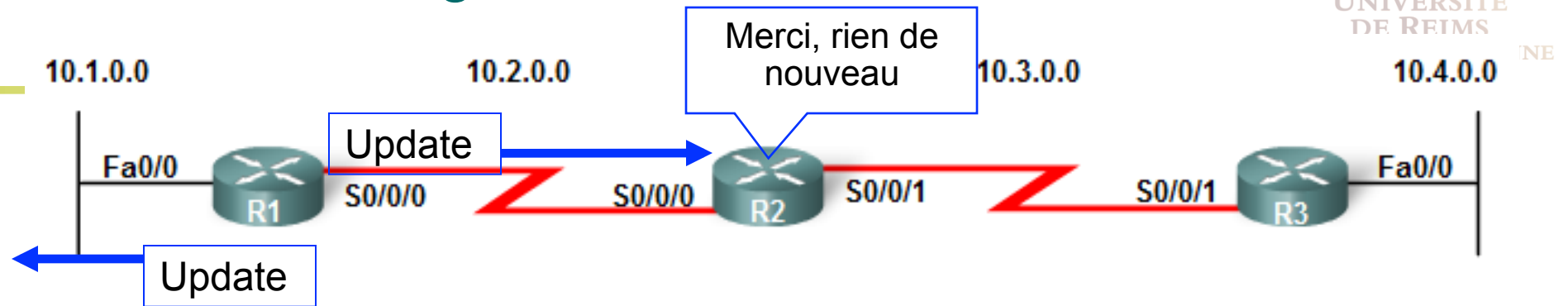
Non

Qu'est-ce qu'il faut apprendre encore ?

R1 ne connaît pas 10.4.0.0

R3 ne connaît pas 10.1.0.0

# Prochain Échange



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
|          |           | !   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/1    | 1   |
|          |           | 2   |

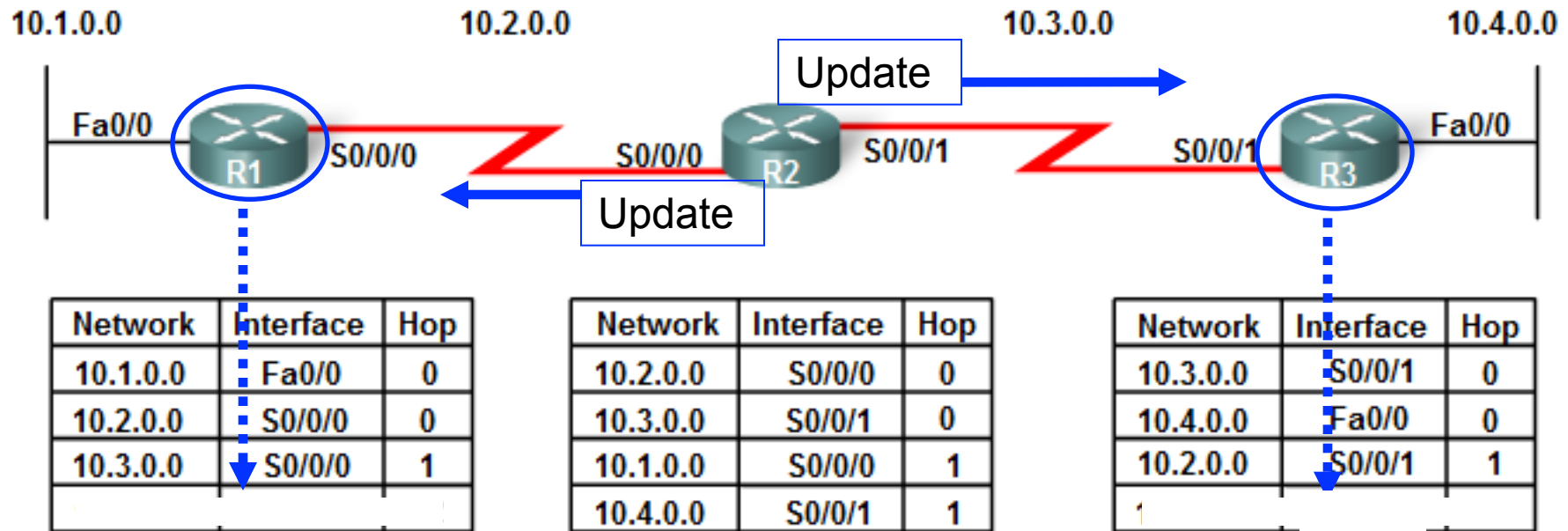
**R1:**

Envoie la totalité de sa table de routage

R2 apprend quelque chose nouvelle ?

Non

# Prochain Échange



**R2:**

Envoie la totalité de sa table de routage

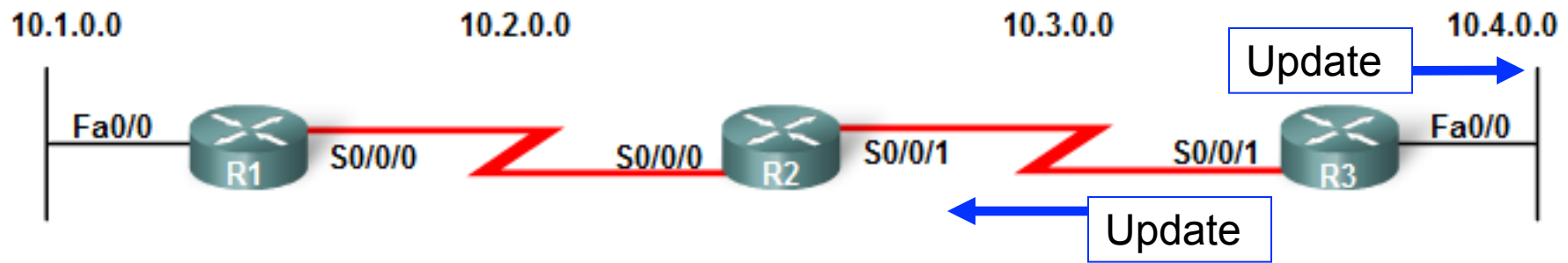
R1 apprend quelque chose nouvelle ?

Oui, 10.4.0.0

R3 apprend quelque chose nouvelle ?

Oui, 10.1.0.0

# Prochain Échange



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/1    | 1   |
| 10.1.0.0 | S0/0/1    | 2   |

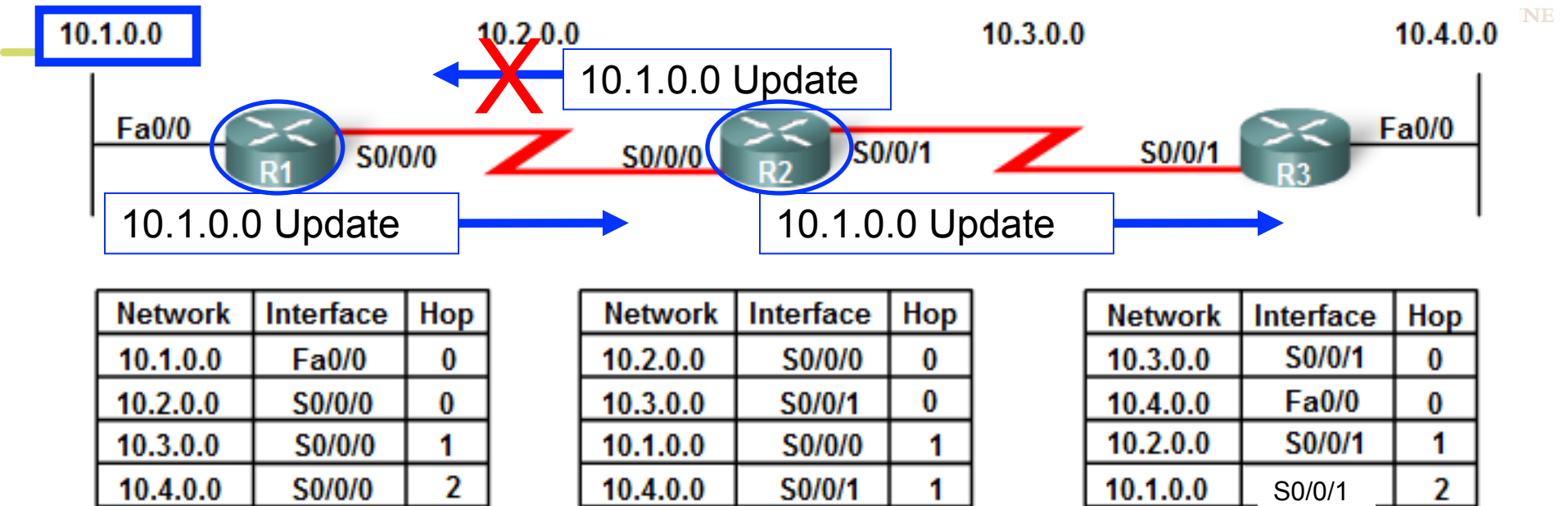
**R3:**

Envoie la totalité de sa table de routage

R2 apprend quelque chose nouvelle ?

Non

# Rappel sur le Split Horizon



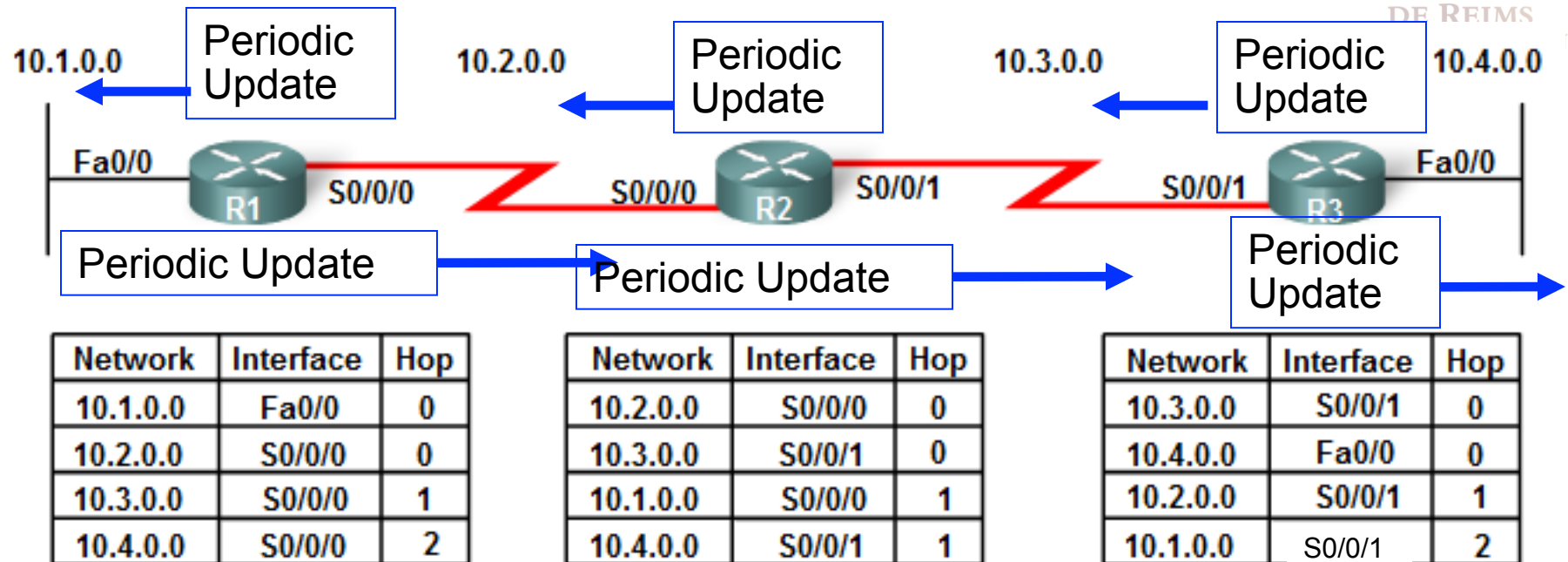
Les protocoles de vecteur de distance généralement implémentent la technique Split Horizon (horizon partagé)

Empêche une information d'être transmise sur la même interface d'où elle a été apprise

# Gestion de la table de routage

- Mises à jour périodiques
  - Triggered Updates

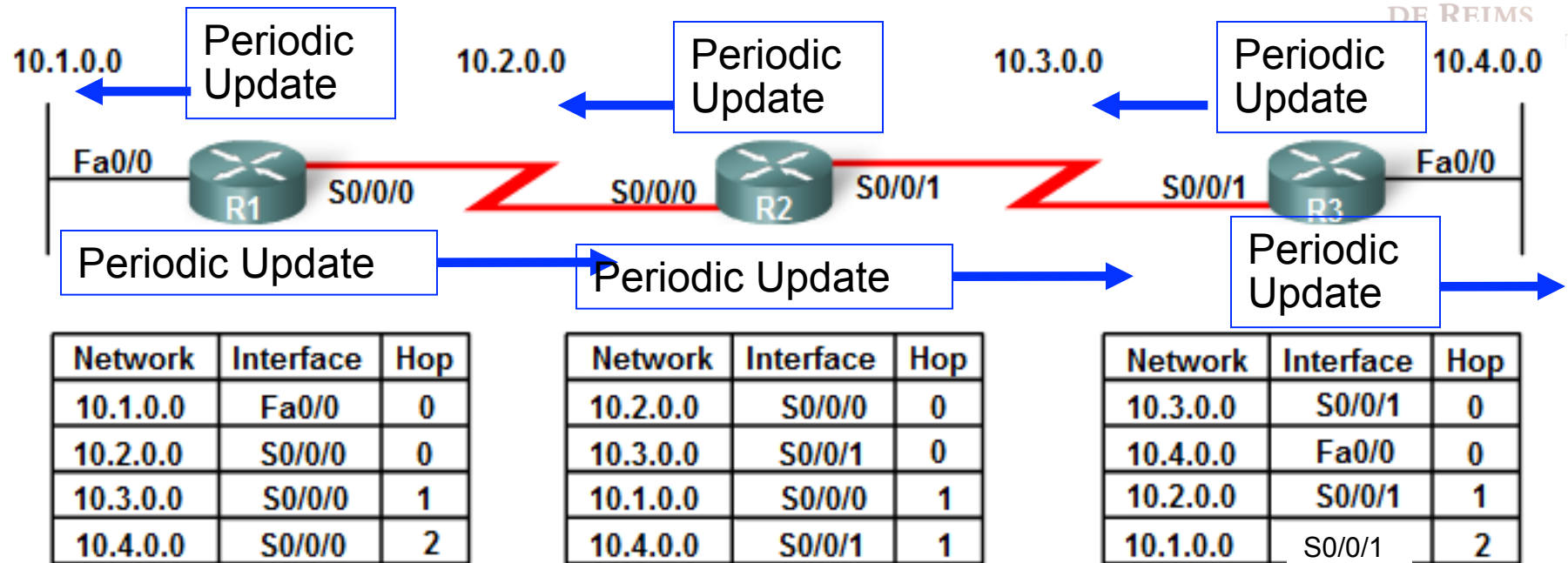
# Mises à jour périodiques



Pour garder les tables de routage à jour, les machines doivent être au courant des modifications grâce à des mises à jour

RIP choisit d'envoyer régulièrement sa table de routage, même quand aucune modification a été observée

# Periodic Updates



Les mises à jour peuvent contenir des informations sur les changements topologiques

Défaillance d'un lien

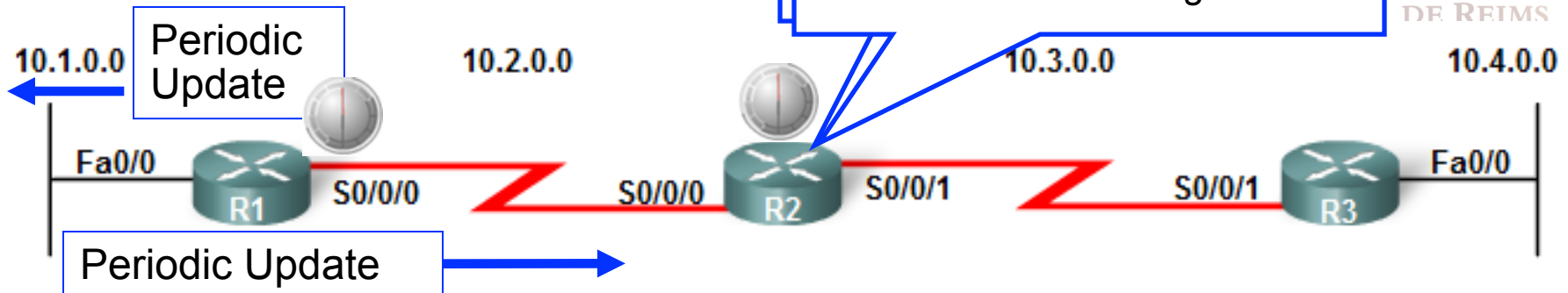
Introduction d'un nouveau lien

Défaillance d'un routeur

Modification des paramètres d'un lien



# Temporisateurs RIP



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/1    | 1   |
| 10.1.0.0 | S0/0/1    | 2   |

**Update timer** : 30 secondes

**Invalid Timer** : si une mise à jour n'est pas reçue au bout de 180 secondes (par défaut), le routeur est marqué comme invalide et sa distance à 16

La route reste dans la table de routage

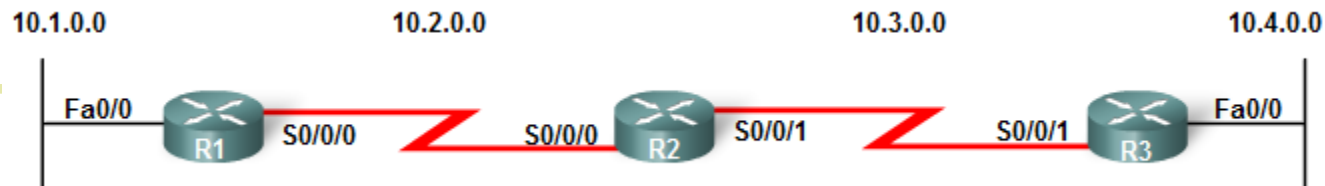
**Flush Timer**: 240 secondes (défaut)

Lorsque le flush time expire, la route est supprimée de la table de routage

**Hold-down Timer**: 180 secondes (défaut)

On verra ça d'ici peu

# Temporisateurs RIP



```
R1# show ip route
```

```

    10.0.0.0/16 is subnetted, 4 subnets
C       10.2.0.0 is directly connected, Serial0/0/0
R       10.3.0.0 [120/1] via 10.2.0.2, 00:00:04, Serial0/0/0
C       10.1.0.0 is directly connected, FastEthernet0/0
R       10.4.0.0 [120/2] via 10.2.0.2, 00:00:04, Serial0/0/0

```

```
R1# show ip protocols
```

```
Routing Protocol is "rip"
```

```
  Sending updates every 30 seconds, next due in 13 seconds
```

```
  Invalid after 180 seconds, hold down 180, flushed after 240
```

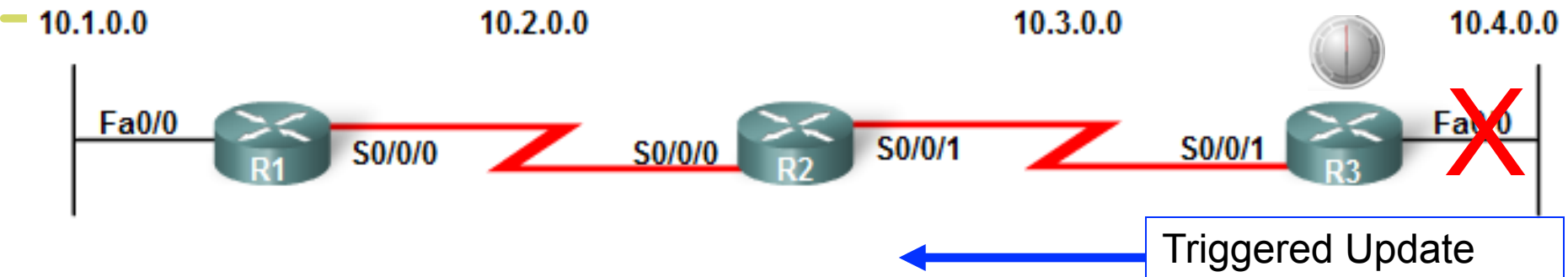
```
<output omitted>
```

```
Routing Information Sources:
```

| Gateway  | Distance | Last Update |
|----------|----------|-------------|
| 10.3.0.1 | 120      | 00:00:27    |

On peut vérifier les timers avec **show ip route** et **show ip protocols**

# Triggered Updates



Un **triggered update** (***mise à jour déclenchée***) est une mise à jour envoyée immédiatement après la réception d'une modification

Pas besoin d'attendre l'intervalle de mise à jours (30 secondes)

Quelle est l'avantage ?

Accélère la convergence

# Boucles de Routage

- Définitions
- Implications
- Conter jusqu'à l'infini
- Prévention grâce à une métrique maximale
  - Prévention avec des Hold-down Timers
  - Prévention avec le Split Horizon

# Définition



Une **boucle de routage** est une condition dans laquelle un paquet est continuellement relayé par une série de routers sans arriver à sa destination

Peut arriver lorsque les informations de routage sont contradictoires/  
incorrectes

Ça n'arrive qu'avec les protocoles à vecteur de distance

La boucle peut être causée par :

Des routes statiques mal configurées

Redistribution incorrecte de routes (CCNP)

Routage contradictoire à cause d'une convergence lente

# Implications

Quels sont les problèmes occasionnés par une boucle de routage ?

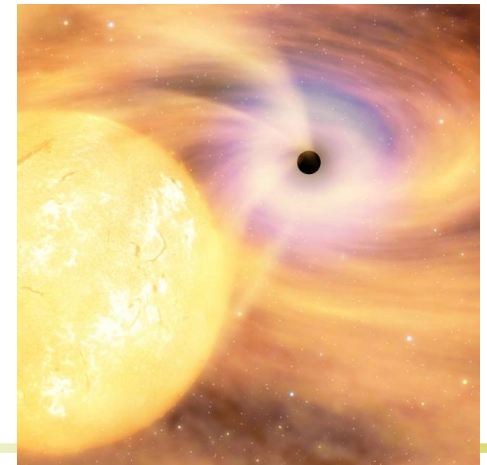
- Surcharge des liens à cause des paquets retransmis

- Surcharge de la CPU du routeur - trop d'effort pour un routage inutile

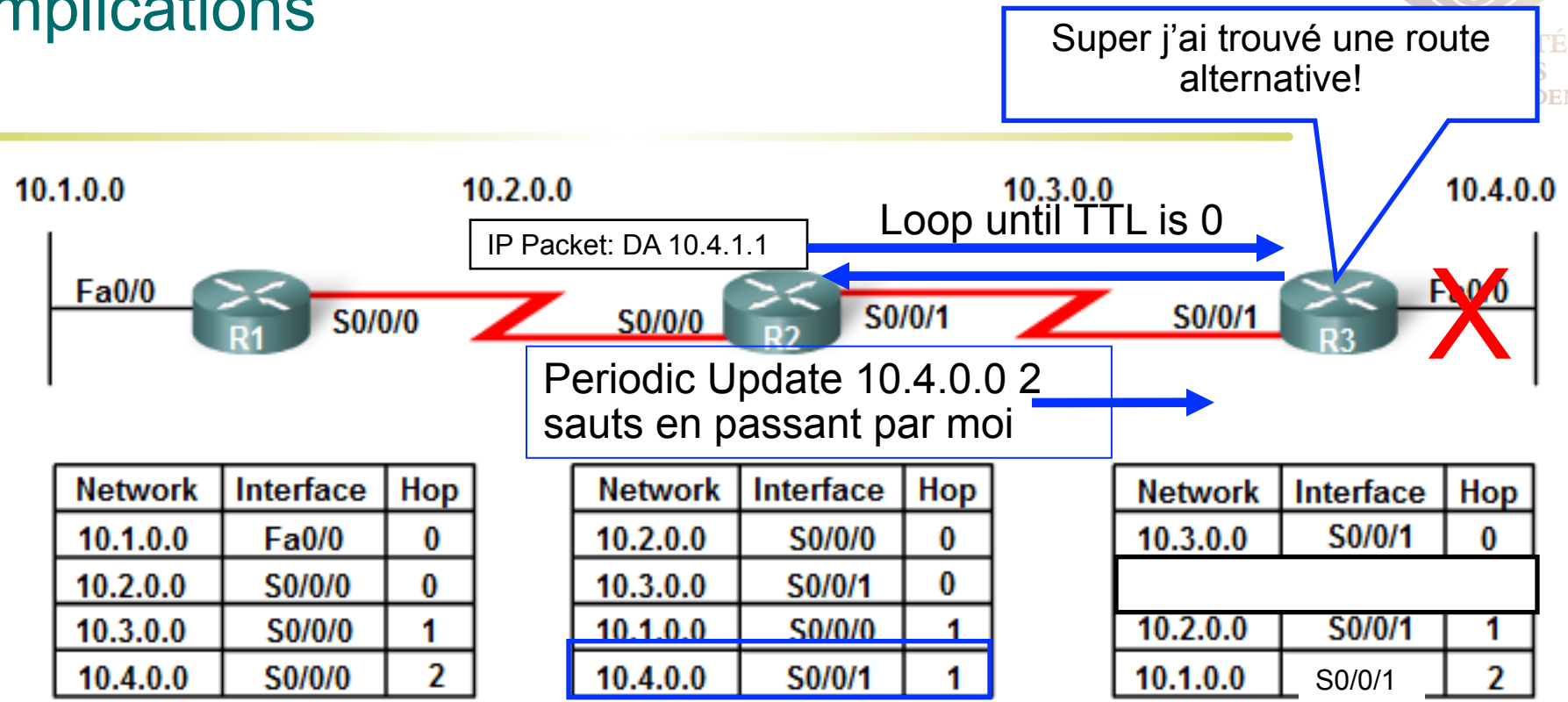
- Des mises à jour peuvent être perdues ou ne pas être traitées à temps

- Les paquets peuvent être perdus dans des “trous noirs”

- Les paquets ne s'arrêtent que lorsque le TTL IP dévient 0



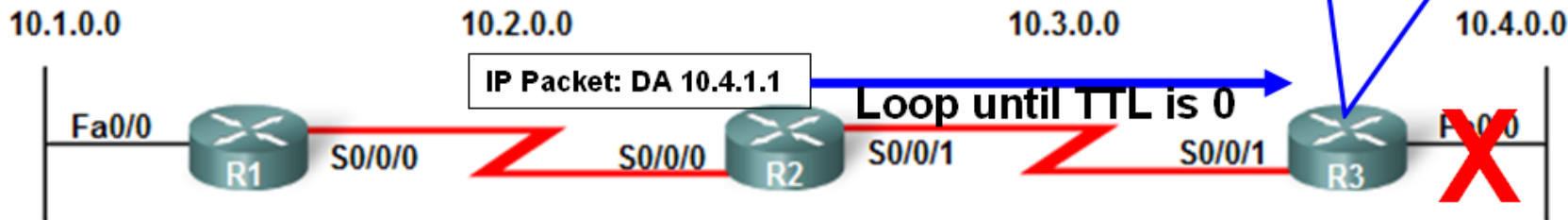
# Implications



Si Split Horizon n'est pas activé, qu'est-ce que arrive si 10.4.0.0 tombe ?

# Implications

J'ai cru par erreur avoir une route vers 10.4.0.0.



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

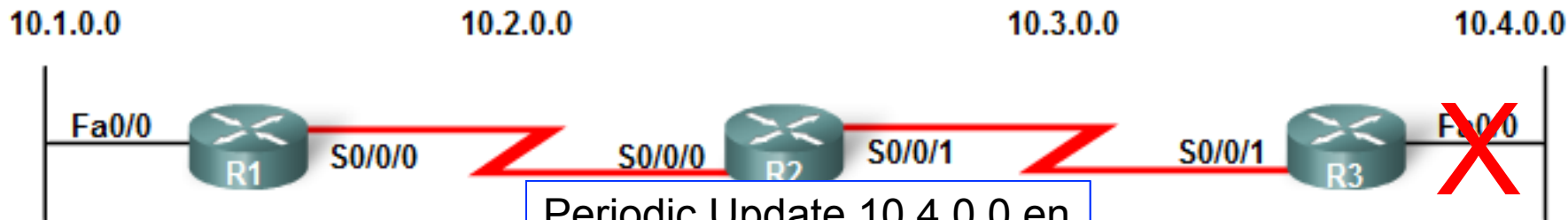
| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | S0/0/1    | 2   |
| 10.2.0.0 | S0/0/1    | 1   |
| 10.1.0.0 | S0/0/1    | 2   |

- Des mécanismes pour éliminer les boucles de routage
  - Définir une valeur maximale pour la métrique (< infini)
  - Utiliser des Hold-down timers
  - Utiliser le Split horizon
  - Faire de la route empoisonnée ou « poison reverse »
  - Des mises à jour déclenchées (déjà vu)



# Compter à l'infini

Periodic Update 10.4.0.0 en 5 sauts en passant par moi



Periodic Update 10.4.0.0 en 4 sauts en passant par moi

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 3   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | S0/0/1    | 4   |
| 10.2.0.0 | S0/0/1    | 1   |
| 10.1.0.0 | S0/0/1    | 2   |

**Compter à l'infini** est une manière de définir une limite maximale pour le nombre de sauts, évitant une boucle éternelle

Chaque protocole a une valeur différente

RIP définit 16 comme « infini »

Une route qui a compté jusqu'à l'infini est marquée comme inatteignable

# Hold-Down Timers

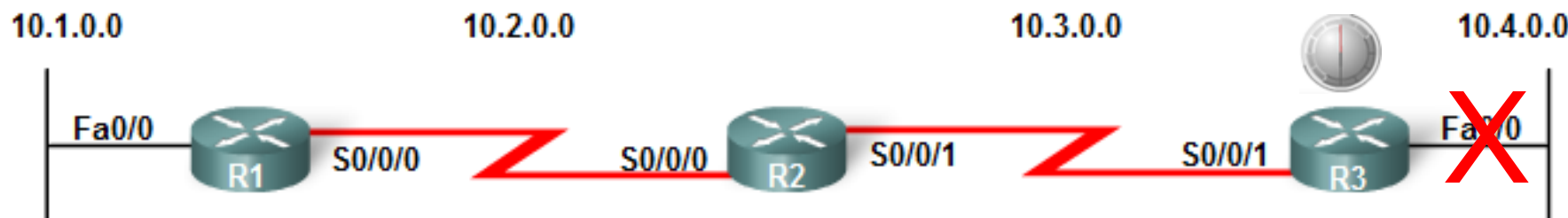


Une boucle de routage peut être créée si une mise à jour est reçue/envoyée en période de instabilité

Hold-down timers :

Empêchent les mises à jour dans les conditions de instabilité

## Hold-Down Timers



Triggered Update

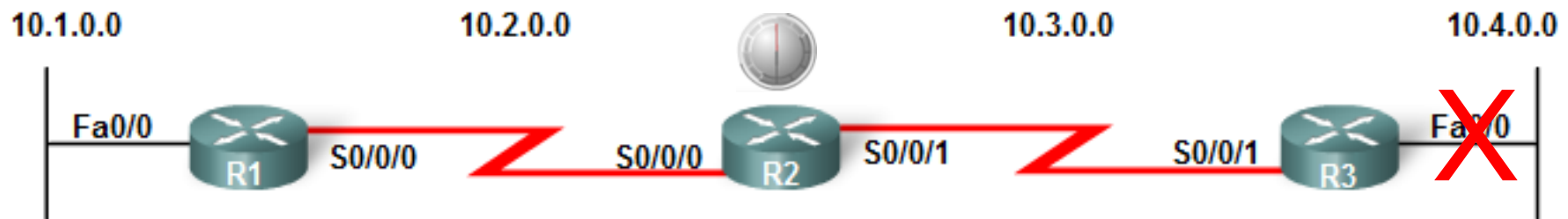
| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network             | Interface        | Hop          |
|---------------------|------------------|--------------|
| 10.3.0.0            | S0/0/1           | 0            |
| <del>10.4.0.0</del> | <del>Fa0/0</del> | <del>0</del> |
| 10.2.0.0            | S0/0/1           | 1            |
| 10.1.0.0            | S0/0/1           | 2            |

Le réseau 10.4.0.0 tombe  
 R3 envoie un triggered update.

# Hold-Down Timers



Triggered Update

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

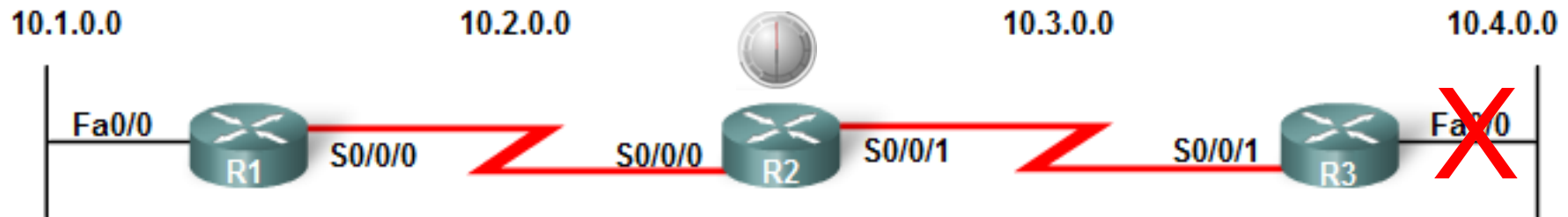
| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network             | Interface        | Hop          |
|---------------------|------------------|--------------|
| 10.3.0.0            | S0/0/1           | 0            |
| <del>10.4.0.0</del> | <del>Fa0/0</del> | <del>0</del> |
| 10.2.0.0            | S0/0/1           | 1            |
| 10.1.0.0            | S0/0/1           | 2            |

Possibly down- Démarrage du Hold-down Timer

R2 reçoit la mise à jour de R3 indiquant que 10.4.0.0 n'est plus accessible  
 R2 marque le réseau comme « possibly down » et démarre le hold-down timer.

# Hold-Down Timers



Triggered Update

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network             | Interface        | Hop          |
|---------------------|------------------|--------------|
| 10.3.0.0            | S0/0/1           | 0            |
| <del>10.4.0.0</del> | <del>Fa0/0</del> | <del>0</del> |
| 10.2.0.0            | S0/0/1           | 1            |
| 10.1.0.0            | S0/0/1           | 2            |

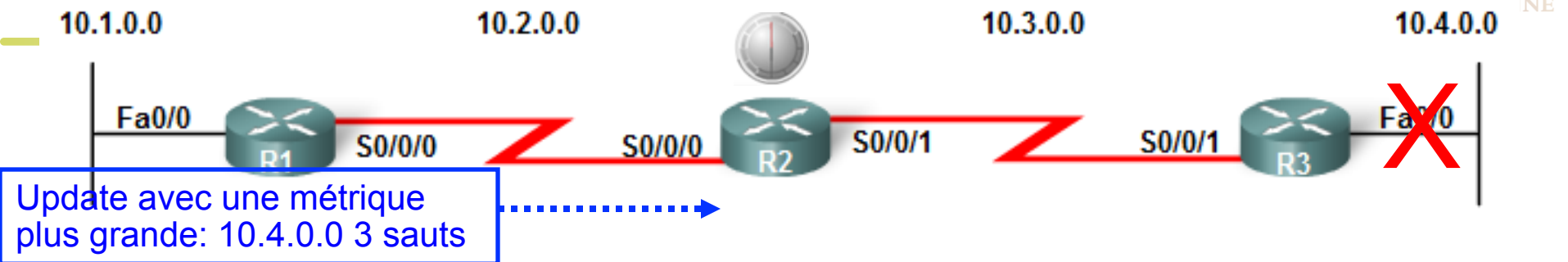
Possibly down - Start Hold-down Timer

Mise à jour avec une  
métrique meilleure

Si une mise à jour avec une **métrique meilleure** est reçue pendant la période d'attente, R2 réactivera le réseau et le hold-down timer sera arrêté

Note : Dans cet exemple la seule meilleure métrique serait 1

## Hold-Down Timers



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

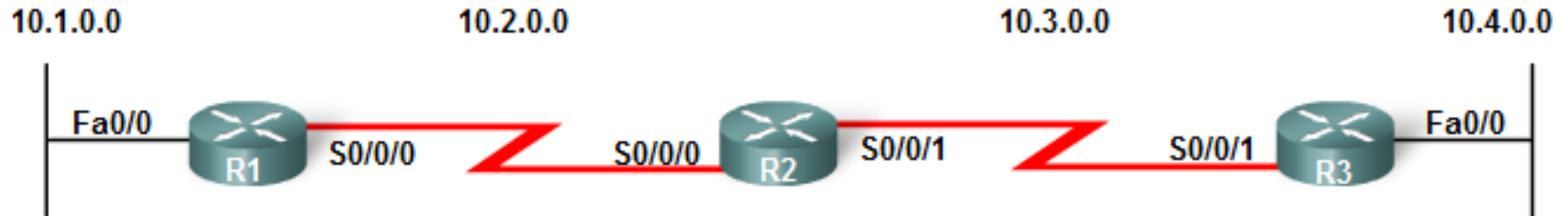
| Network             | Interface        | Hop          |
|---------------------|------------------|--------------|
| 10.3.0.0            | S0/0/1           | 0            |
| <del>10.4.0.0</del> | <del>Fa0/0</del> | <del>0</del> |
| 10.2.0.0            | S0/0/1           | 1            |
| 10.1.0.0            | S0/0/1           | 2            |

Still possibly down - Keep Hold-down Timer going

Si une mise à jour avec une **métrique plus grande** est reçue pendant la période d'attente, ignore cette mise à jour

Ça donne le temps à la propagation de l'information de R3

# Split Horizon



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

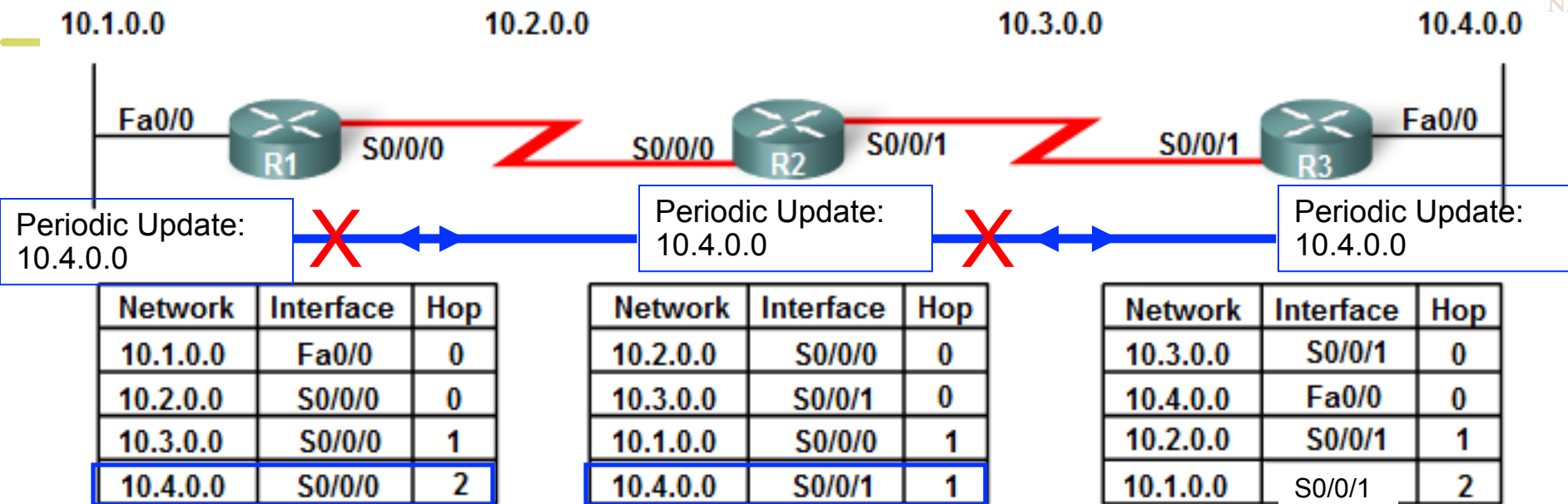
| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/1    | 1   |
| 10.1.0.0 | S0/0/1    | 2   |

**Split horizon** empêche une route d'être annoncée dans la direction d'où elle a été apprise

Empêche que les routeurs soient « réalimentés » avec des informations anciennes

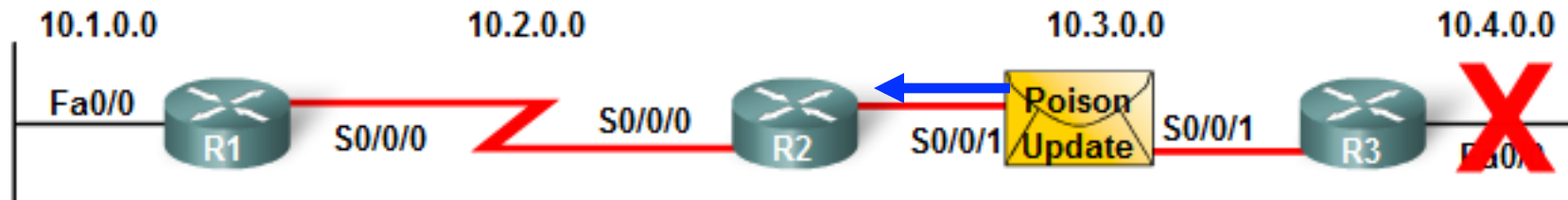
# Split Horizon



1. R3 annonce le réseau 10.4.0.0 à R2.
2. R2 reçoit l'information et met à jour sa table de routage.
3. R2 annonce 10.4.0.0 à R1 via S0/0/0.  
R2 n'annonce pas 10.4.0.0 à R3 via S0/0/1, car cette route est issue de cette interface.
4. R1 reçoit l'information et met à jour sa table de routage.
5. À cause du split horizon, R1 n'annonce pas 10.4.0.0 sur R2 non plus.



# Route empoisonnée



| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.1.0.0 | Fa0/0     | 0   |
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/0    | 2   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0    | 0   |
| 10.3.0.0 | S0/0/1    | 0   |
| 10.1.0.0 | S0/0/0    | 1   |
| 10.4.0.0 | S0/0/1    | 1   |

| Network  | Interface | Hop |
|----------|-----------|-----|
| 10.3.0.0 | S0/0/1    | 0   |
| 10.4.0.0 | Fa0/0     | 16  |
| 10.2.0.0 | S0/0/1    | 1   |
| 10.1.0.0 | S0/0/1    | 2   |

C'est une technique pour ***marquer une route comme inatteignable*** lors d'une mise à jour envoyée aux autres machines.

**Inatteignable** = métrique au maximum.

Les routes empoisonnées accélèrent la convergence

Cette technique marche très bien en tandem avec le Split Horizon

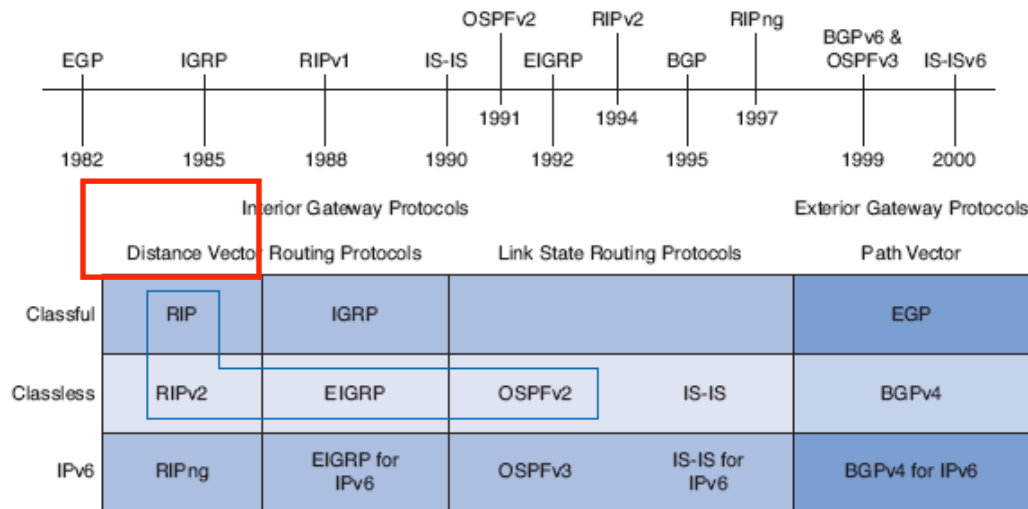
# RIPv1: Un protocole de routage classful à vecteur de distance

---

- Historique et Perspectives
- Caractéristiques de RIPv1
- Utilisation de RIP

# RIPv1: protocole classful à vecteur de distance

- ▶ RIP (Routing Information Protocol) fut le premier protocole utilisé.
- ▶ Un protocole encore très populaire : simple et facile à entretenir.
- ▶ Pourquoi étudier RIP ?
  - Encore en utilisation (il y a RIPvng pour IPv6).
  - Permet de comprendre les mécanismes des protocoles dynamiques
    - versions classful (RIPv1) et classless (RIPv2)



Highlighted routing protocols are the focus of this course.

# Caractéristiques et Format des Messages

## ► Note :

- La première version de RIP est appelée RIPv1 pour la différencier de RIP version 2 (RIPv2).
- Les deux versions partagent plusieurs caractéristiques. Si les éléments sont communs nous parlerons de RIP. Si les éléments sont spécifiques, nous utiliserons RIPv1 ou RIPv2.

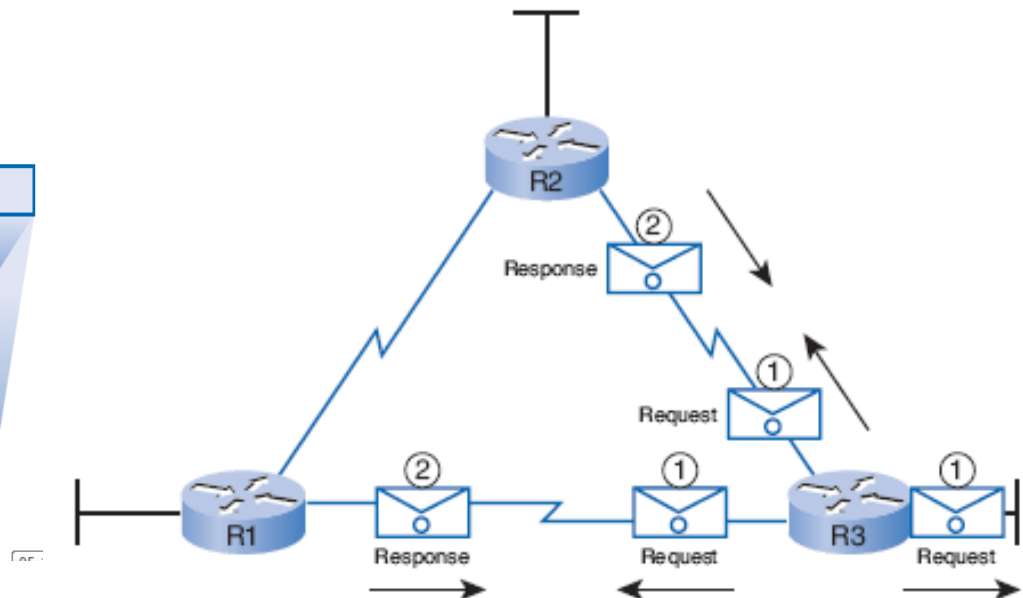
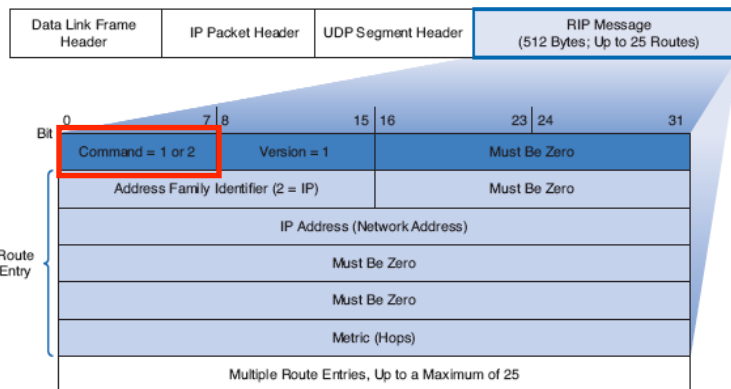
## ► Caractéristiques de RIP :

- Protocole de routage à vecteur de distance
- Métrique : sauts (hop count)
- Les routes annoncées avec un hop count supérieur à 15 sont considérées inatteignables (unreachable).
- Les messages (mises à jour des tables de routage ) sont broadcastés à chaque 30 secondes. (RIPv2 utilise multicast)

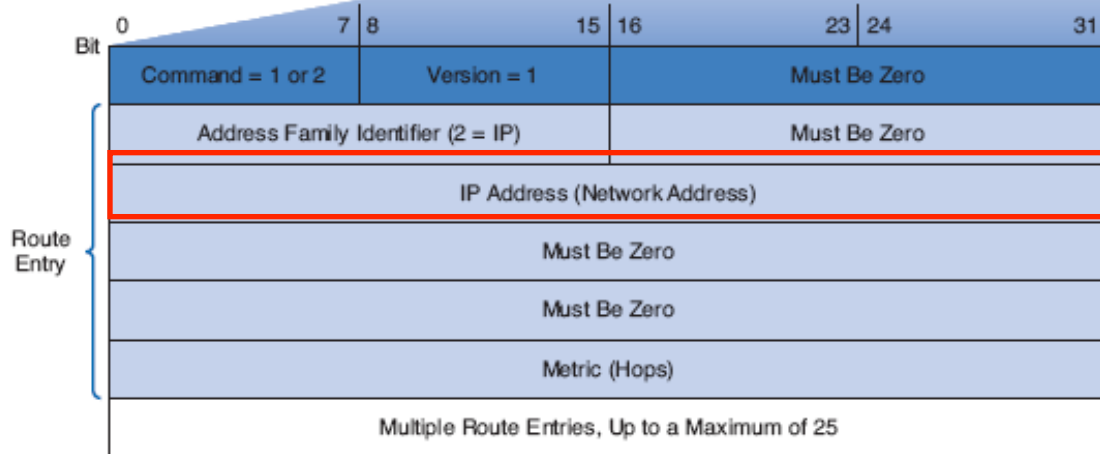
# Fonctionnement de RIP

## Initialisation

1. Chaque interface RIP envoie un message **Request**
    - Demande des tables de routage entières.
  2. Un message **Response** est envoyé par les voisins RIP.
    - Si nouvelle route : l'installe dans les tables de routage.
    - Si une route existante : la remplace si le hop count est plus petit.
- Périodiquement envoie des **triggered update** pour mettre à jour les voisins.



# Routage classful



Class A Address Range: 0.0.0.0 to 126.255.255.255  
 Class B Address Range: 128.0.0.0 to 191.255.255.255  
 Class C Address Range: 192.0.0.0 to 223.255.255.255

Aucun  
masque  
envoyé

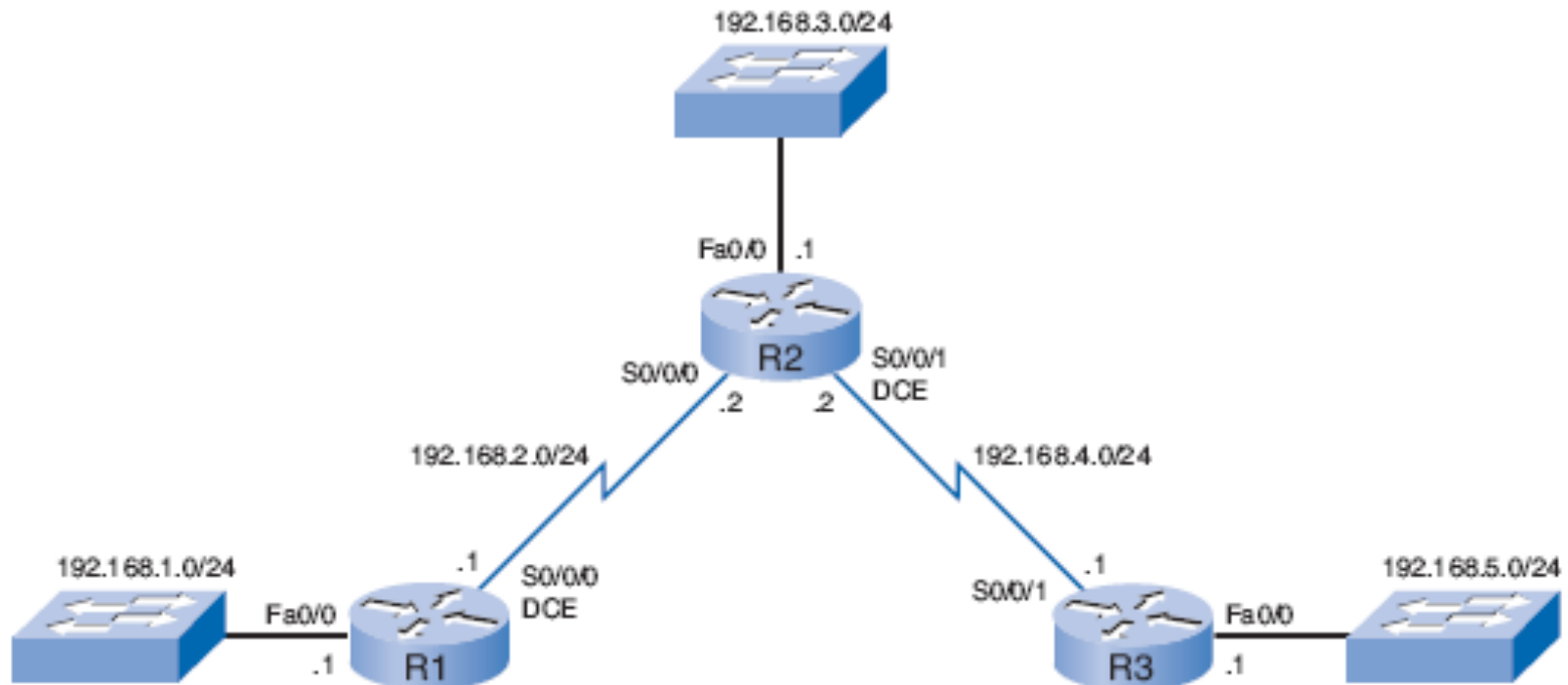
# Configuration de base RIPv1

---

- RIPv1 Scénario A
- Activer RIP: la commande router rip
- Specification des réseaux

# RIPv1 Scénario A

- ▶ Cette topologie utilise cinq réseaux Classe C.
- ▶ RIPv1 est un protocole classful
- ▶ La classe des réseaux est utilisé par RIPv1 pour déterminer le masque.





# Activation de RIP : la commande router rip

- **router rip** entre dans le mode de configuration RIP.

```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# router ?
  bgp          Border Gateway Protocol (BGP) ^
  egp          Exterior Gateway Protocol (EGP) ^
  eigrp        Enhanced Interior Gateway Routing Protocol (EIGRP) ^
  igrp         Interior Gateway Routing Protocol (IGRP) ^
  isis         ISO IS-IS
  iso-igrp     IGRP for OSI networks
  mobile       Mobile routes
  odr          On Demand stub Routes
  ospf         Open Shortest Path First (OSPF) ^
  rip          Routing Information Protocol (RIP) ^
R1(config)# router rip
R1(config-router)#
```

# Activation de RIP : la commande `router rip`

- ▶ **`router rip`**
  - N'active pas directement le protocole.
  - Permet d'accéder à l'environnement de configuration.
- ▶ Tant que le protocole n'est pas configuré aucun message sera envoyé.
- ▶ **`no router rip`**
  - Utilisé pour arrêter le processus RIP
  - Efface toute configuration RIP existante.

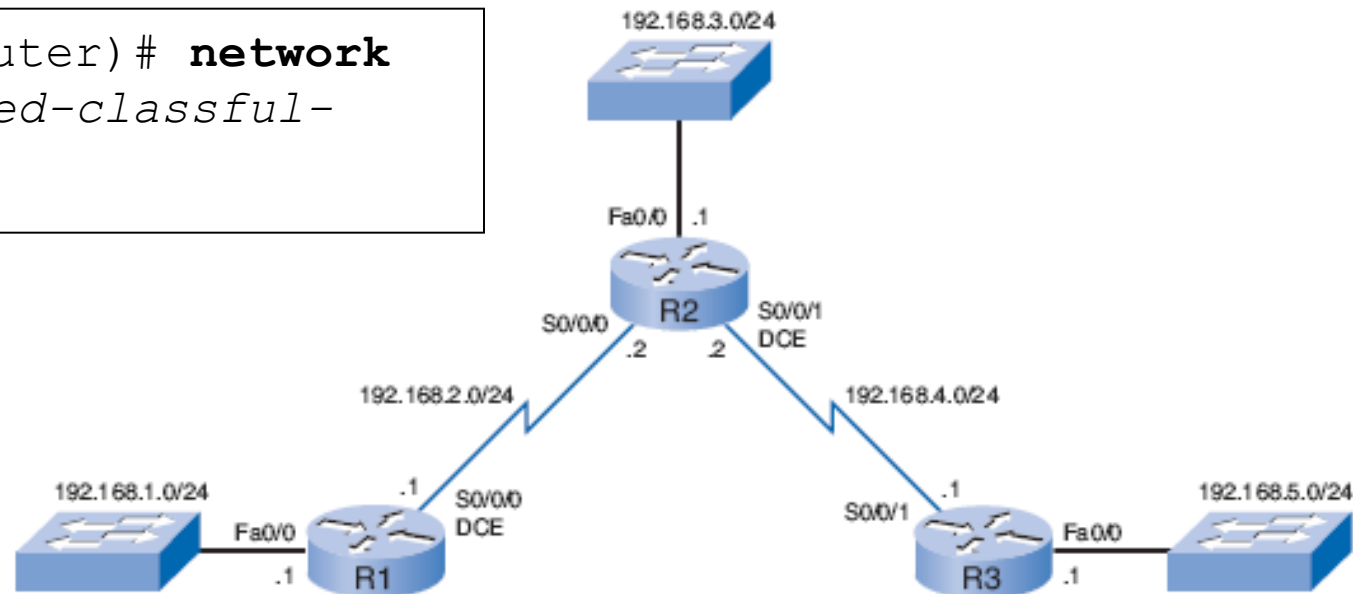
```
R1# conf t  
R1(config)# router rip  
R1(config-router)#
```

- ▶ Pour activer RIP dans un réseau, utiliser la commande **network** dans le mode de configuration du routeur
- ▶ Entrer directement l'adresse classful du réseau concerné.

```
Router(config-router)# network directly-connected-classful-  
network-address
```

# Spécification des réseaux

```
Router(config-router)# network  
directly-connected-classful-  
network-address
```



## ► La commande **network** a la fonction de :

- Activer RIP sur toutes les interfaces appartenant à un réseau.
  - Les interfaces associées peuvent désormais envoyer des messages.
- Propager le réseau spécifié dans les updates RIP envoyés aux autres routeurs à chaque 30 secondes.

# Spécification des réseaux

- ▶ Si on donne une adresse machine ou de sous-réseau, IOS convertit automatiquement vers l'adresse classful.
- ▶ Par exemple, si on entre **network 192.168.1.32**, le routeur le convertit en **network 192.168.1.0**.

```
R1(config)# router rip
```

```
R1(config-router)# network 192.168.1.0
```

```
R1(config-router)# network 192.168.2.0
```

Uniquement les réseaux classful  
directement connectés

```
R2(config)# router rip
```

```
R2(config-router)# network 192.168.2.0
```

```
R2(config-router)# network 192.168.3.0
```

```
R2(config-router)# network 192.168.4.0
```

```
R3(config)# router rip
```

```
R3(config-router)# network 192.168.4.0
```

```
R3(config-router)# network 192.168.5.0
```

# Vérification et Dépannage

---

- Verification RIP : show ip route
- Verification RIP: show ip protocols
- Verification RIP: debug ip rip
- Interfaces passives

# Verification RIP: show ip route

- ▶ Un **R** indique une route apprise via RIP.
- ▶ Normalement utilisé pour vérifier la convergence car liste toutes les routes (directement connectées, statiques, dynamiques)
- ▶ Les routes peuvent ne pas être affichées car le protocole peut tarder à converger

```
R1# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile,  
<output omitted>
```

```
Gateway of last resort is not set
```

```
R    192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:02, Serial0/0/0  
R    192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:02, Serial0/0/0  
C    192.168.1.0/24 is directly connected, FastEthernet0/0  
C    192.168.2.0/24 is directly connected, Serial0/0/0  
R    192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:02, Serial0/0/0
```

# Verification RIP: show ip route

```
R1# show ip route
```

```
<output omitted>
```

```
R      192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:23, Serial0/0/0
```

| Output           | Description                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------|
| R                | Identifies the source of the route as RIP.                                                                       |
| 192.168.5.0      | Indicates the address of the remote network.                                                                     |
| /24              | Indicates the subnet mask used for this network.                                                                 |
| [120/2]          | Shows the administrative distance (120) and the metric (2 hops).                                                 |
| via 192.168.2.2, | Specifies the address of the next-hop router (R2) to send traffic to for the remote network.                     |
| 00:00:23,        | Specifies the amount of time since the route was updated (here, 23 seconds). Another update is due in 7 seconds. |
| Serial0/0/0      | Specifies the local interface through which the remote network can be reached.                                   |



# Verification RIP: show ip protocols

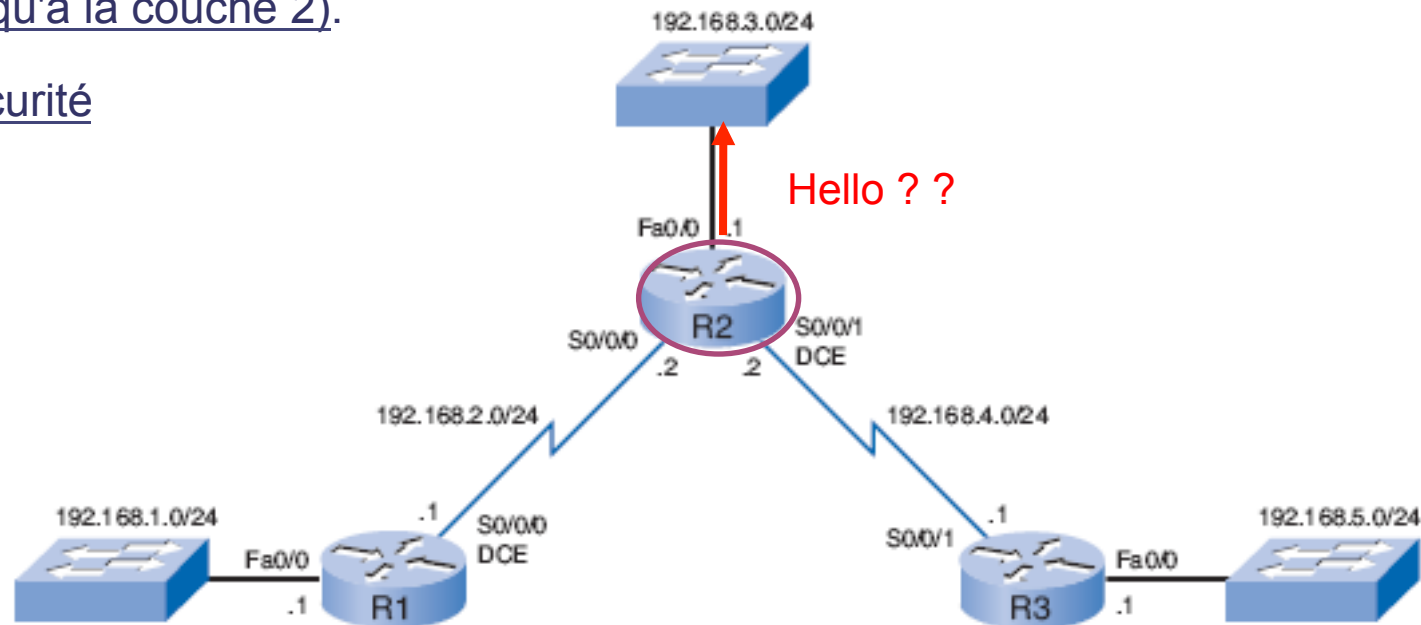
- ▶ Vérifie que RIP est active et configuré sur le routeur R2
- ▶ Au moins une interface associée avec la commande **network** est nécessaire

```

R2#show ip protocols
① Routing Protocol is "rip"
② { Sending updates every 30 seconds, next due in 23 seconds
      Invalid after 180 seconds, hold down 180, flushed after 240
      Outgoing update filter list for all interfaces is not set
      Incoming update filter list for all interfaces is not set
      Redistributing: rip
      Default version control: send version 1, receive any version
      Interface          Send  Recv  Triggered RIP  Key-chain
④ { FastEthernet0/0      1     1 2
      Serial0/0/0        1     1 2
      Serial0/0/1        1     1 2
      Automatic network summarization is in effect
      Maximum path: 4
      Routing for Networks:
⑥ { 192.168.2.0
      192.168.3.0
      192.168.4.0
      Routing Information Sources:
      Gateway            Distance    Last Update
⑦ { 192.168.2.1           120        00:00:18
      192.168.4.1           120        00:00:22
      Distance: (default is 120)
  
```

# Interfaces Passives

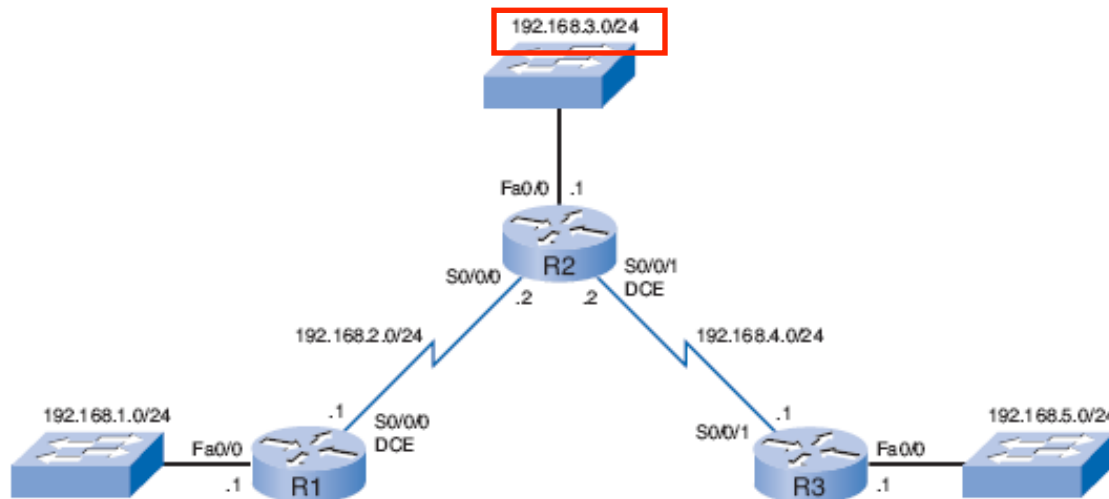
- ▶ Certains routeurs ont des interfaces qui ne se connectent pas à d'autres routeurs exécutant RIP.
- ▶ La commande **passive-interface** empêche RIP d'envoyer des updates sur ces interfaces.
  - Évite le gaspillage de la bande passante.
  - Les dispositifs dans la LAN sont surchargés avec le traitement de messages RIPv1 (au moins jusqu'à la couche 2).
  - Risque de sécurité



# Interfaces Passives

- ▶ Si nous utilisons  
R2 (Config-router) # **no network 192.168.3.0**
- ▶ R2 n'annonce plus ce LAN dans les updates envoyés à R1 et R3.
- ▶ La solution est d'utiliser la commande **passive-interface**

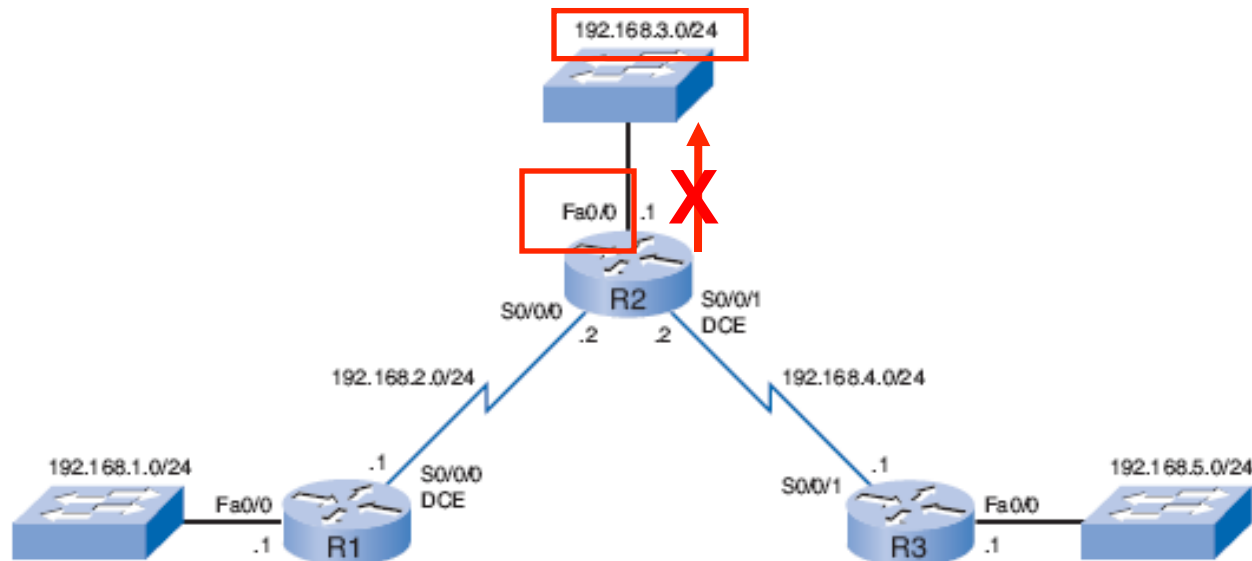
```
Router(config-router) # passive-interface interface-type interface-number
```



# Interfaces Passives

```
R2(config)# router rip
```

```
R2(config-router)# passive-interface FastEthernet 0/0
```



# Interfaces Passives

```
R2# show ip protocols
```

```
<output omitted>
```

| Interface   | Send | Recv | Triggered RIP Key-chain |
|-------------|------|------|-------------------------|
| Serial0/0/0 | 1    | 1 2  |                         |
| Serial0/0/1 | 1    | 1 2  |                         |

*FastEthernet 0/0 no longer  
included*

```
Automatic network summarization is in effect
```

```
Routing for Networks:
```

|                    |                                                                    |
|--------------------|--------------------------------------------------------------------|
| 192.168.2.0        |                                                                    |
| <b>192.168.3.0</b> | <i>LAN network still included in<br/>RIP updates that are sent</i> |
| 192.168.4.0        |                                                                    |

**Passive Interface(s):** *FastEthernet 0/0 is a passive interface*  
**FastEthernet0/0**

```
Routing Information Sources:
```

| Gateway     | Distance | Last Update |
|-------------|----------|-------------|
| 192.168.2.1 | 120      | 00:00:27    |
| 192.168.4.1 | 120      | 00:00:23    |

```
Distance: (default is 120) ^
```

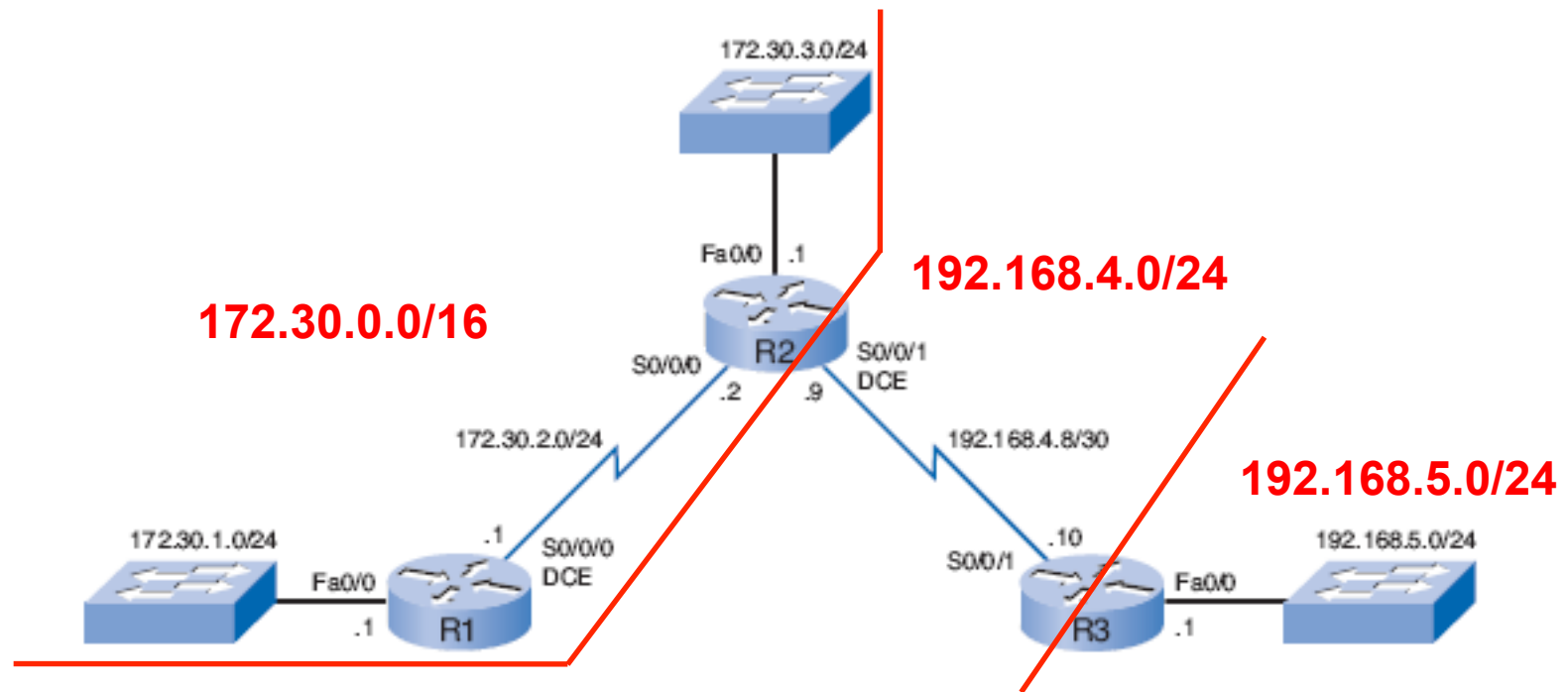
# Agrégation Automatique

---

- Topologie B
- Routeurs frontaliers et l'Agrégation Automatique
- Le traitement des updates RIP
- L'envoi d'updates RIP
- Avantages et Inconvénients de l'Agrégation Automatique

# Une autre topologie : Scénario B

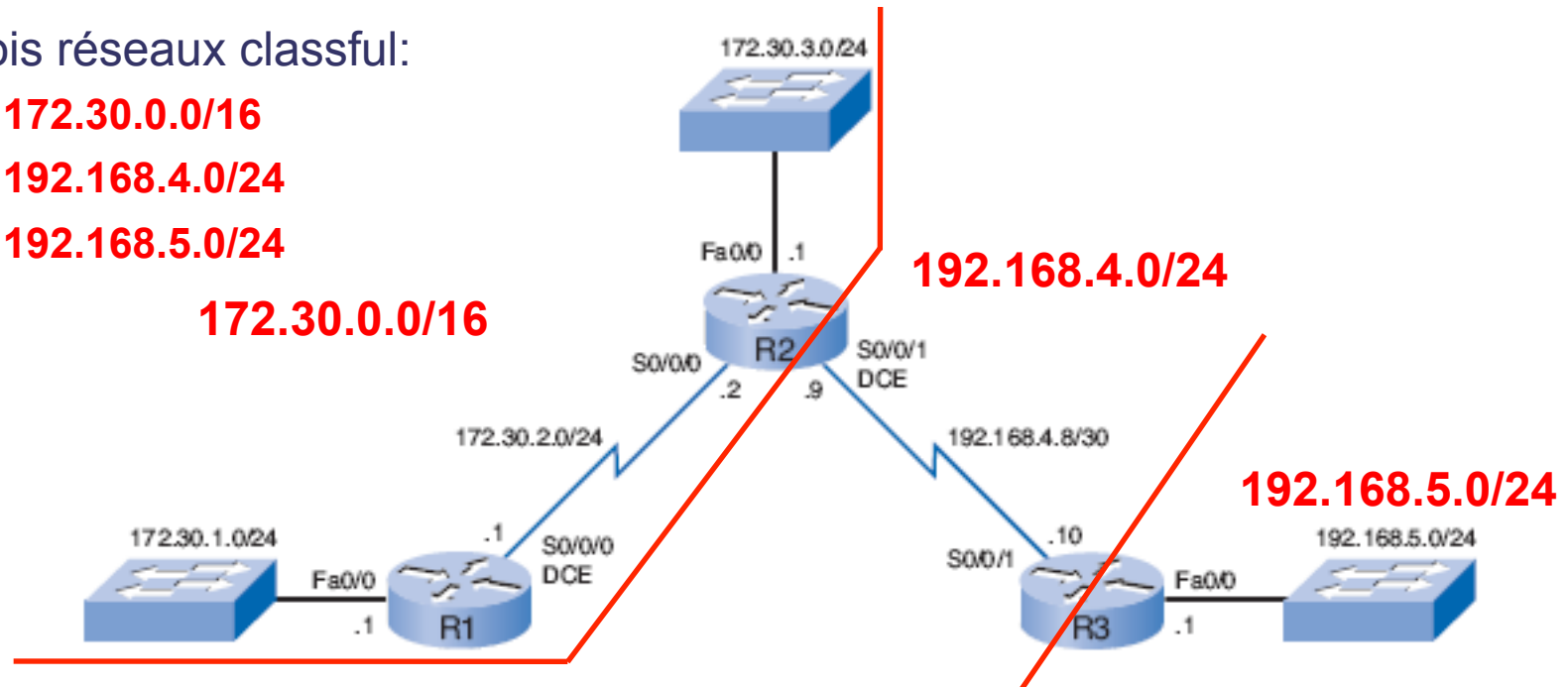
- ▶ La réduction du nombre d'entrées dans la table de routage augmente la performance lors de la recherche des matchings.
- ▶ Certains protocoles de routage, comme RIP, agrègent automatiquement les routes dans certains routeurs.



# Une autre topologie : Scénario B

► Trois réseaux classful:

- **172.30.0.0/16**
- **192.168.4.0/24**
- **192.168.5.0/24**



- Le réseau **172.30.0.0/16** est subnetté en :
  - 172.30.1.0/24
  - 172.30.2.0/24
  - 172.30.3.0/24
- Le réseau **192.168.4.0/24** est subnetté en 192.168.4.8/30.



# Modifiant la configuration de R1

```
R1(config)# interface fa0/0
R1(config-if)# ip address 172.30.1.1 255.255.255.0
R1(config-if)# interface S0/0/0
R1(config-if)# ip address 172.30.2.1 255.255.255.0
R1(config-if)# no router rip
R1(config)# router rip
R1(config-router)# network 172.30.1.0
R1(config-router)# network 172.30.2.0
R1(config-router)# passive-interface FastEthernet 0/0
R1(config-router)# end
R1# show run
<output omitted>
!
router rip
passive-interface FastEthernet0/0
network 172.30.0.0
```

Les commandes **no shutdown** et **clock rate** ne sont pas nécessaires car déjà entrés en Scénario A.

*IOS automatiquement transforme les réseaux en classful*

# Modifiant la configuration de R2

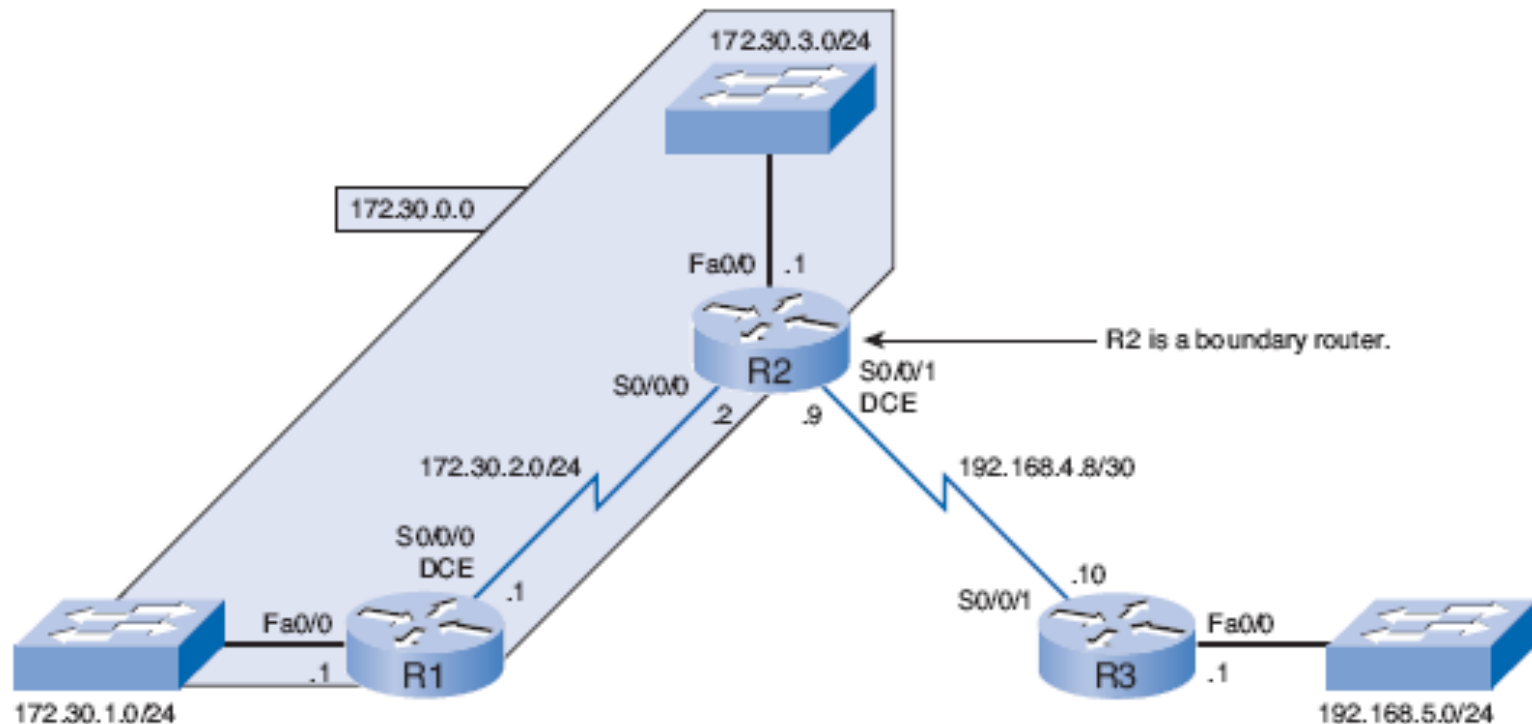
```
R2(config)# interface S0/0/0
R2(config-if)# ip address 172.30.2.2 255.255.255.0
R2(config-if)# interface fa0/0
R2(config-if)# ip address 172.30.3.1 255.255.255.0
R2(config-if)# interface S0/0/1
R2(config-if)# ip address 192.168.4.9 255.255.255.252
R2(config-if)# no router rip
R2(config)# router rip
R2(config-router)# network 172.30.0.0
R2(config-router)# network 192.168.4.8
R2(config-router)# passive-interface FastEthernet 0/0
R2(config-router)# end
R2# show run
router rip
passive-interface FastEthernet0/0
network 172.30.0.0
network 192.168.4.0
```

# Modifiant la configuration de R3

```
R3(config)# interface fa0/0
R3(config-if)# ip address 192.168.5.1 255.255.255.0
R3(config-if)# interface S0/0/1
R3(config-if)# ip address 192.168.4.10 255.255.255.252
R3(config-if)# no router rip
R3(config)# router rip
R3(config-router)# network 192.168.4.0
R3(config-router)# network 192.168.5.0
R3(config-router)# passive-interface FastEthernet 0/0
R3(config-router)# end
R3# show run
<output omitted>
!
router rip
passive-interface FastEthernet0/0
network 192.168.4.0
network 192.168.5.0
!
<output omitted>
```

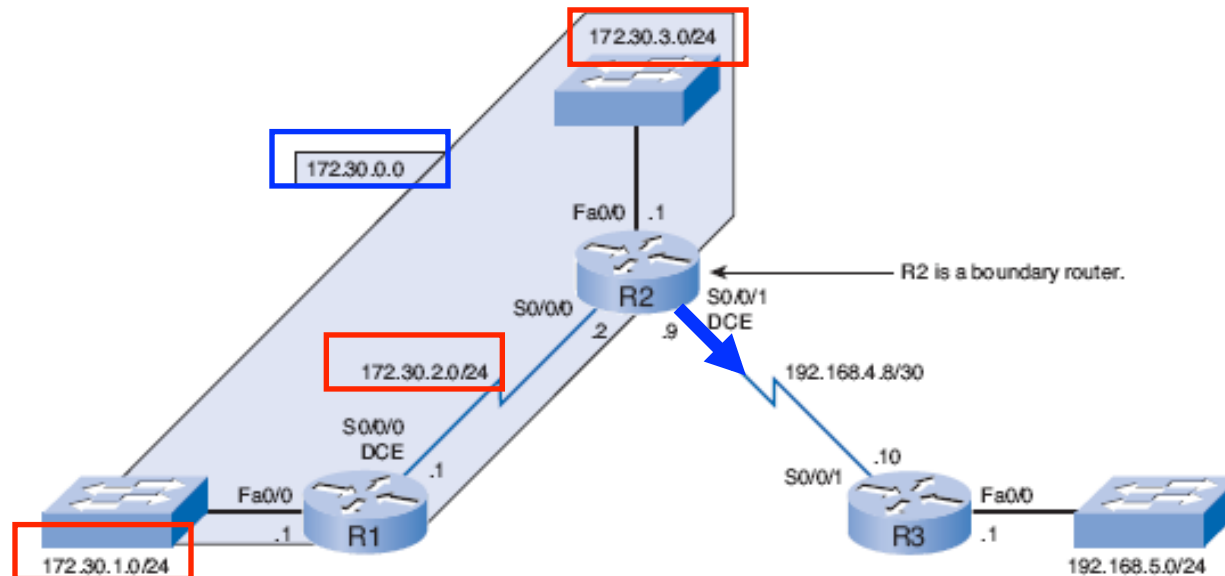
# Routeurs Frontaliers et l'Agrégation

- **RIP** est un protocole classful qui *agrège automatiquement les réseaux classful lors du passage entre des "frontières" d'adresses.*

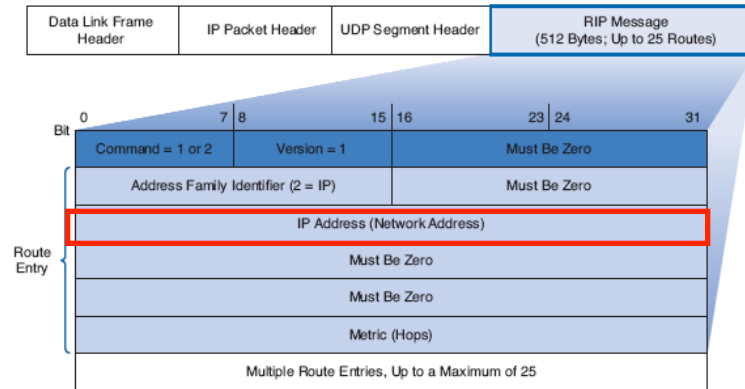


# Routeurs Frontaliers et l'Agrégation

- ▶ R2 a des interfaces sur plusieurs réseaux classful.
- ▶ Cela fait de R2 un routeur frontalier (**boundary router**) dans RIP.
  - Serial 0/0/0 et FastEthernet 0/0 de R2 sont à l'intérieur de la zone 172.30.0.0.
  - Serial 0/0/1 est à l'intérieur de la zone 192.168.4.0.
- ▶ Les routeurs frontaliers regroupent les sous-réseaux RIP lors du passage des messages d'update.
- ▶ Les updates pour **172.30.1.0, 172.30.2.0 et 172.30.3.0** seront automatiquement annoncés comme **172.30.0.0** quand envoyés par l'interface Serial 0/0/1 de R2.



# Traitement des Updates RIP



- ▶ Les protocoles classful n'incluent pas le masque dans les updates.
- ▶ Cependant, la table de routage inclut des routes RIPv1 avec l'adresse réseau et le masque.
- ▶ *Alors comment déterminer le masque qui sera rajouté à une route dans la table de routage ?*

```
R2# show ip route
    172.30.0.0/24 is subnetted, 3 subnets
R       172.30.1.0 [120/1] via 172.30.2.1, 00:00:18, Serial0/0/0
C       172.30.2.0 is directly connected, Serial0/0/0
C       172.30.3.0 is directly connected, FastEthernet0/0
    192.168.4.0/30 is subnetted, 1 subnets
C       192.168.4.8 is directly connected, Serial0/0/1
R       192.168.5.0/24 [120/1] via 192.168.4.10, 00:00:16, Serial0/0/1
```

# Règles pour le traitement des updates RIPv1

- ▶ Les deux règles suivantes déterminent le fonctionnement de RIPv1:
  - Si un **update** et l'**interface** dont il est issu appartiennent au même **grand réseau**, le **masque de l'interface** est utilisé pour les routes apprises.
  - ▶ Si un **update** et l'**interface** dont il est issu appartiennent à différents **grands réseaux**, le **masque classful de l'adresse** est utilisé pour les routes apprises.

## Routing Update and Interface

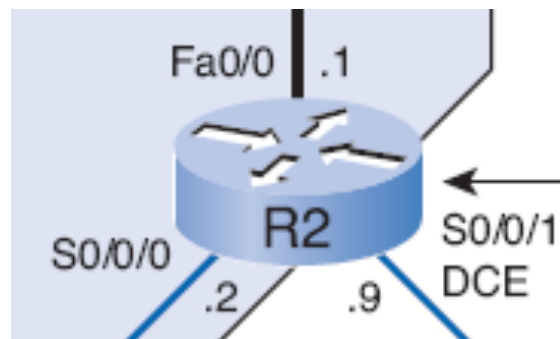
Same Classful Major Network .....

Different Classful Major Network .....

## Routing Update Subnet Mask

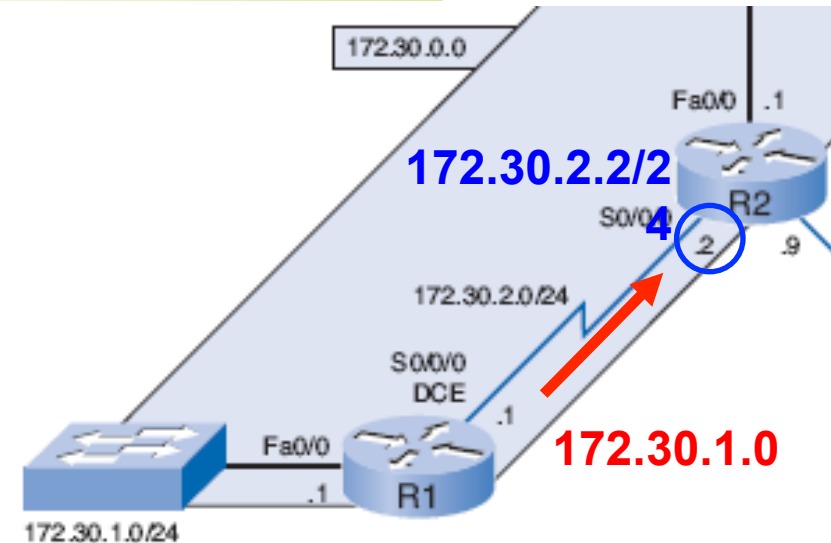
Use mask of interface

Use default classful mask



# Exemple du traitement d'un update RIPv1

- ▶ **Même réseau classful** que l'update arrivant.
- ▶ Update: **172.30.1.0** en 1 saut
- ▶ Interface d'entrée :
  - Serial 0/0/0 - **172.30.2.2/24**
- ▶ **Même réseau classful (172.30.0.x)**
- ▶ Appliquer le masque de S0/0/0 interface, /24.
- ▶ Le sous-réseau 172.30.1.0 /24 est rajouté à la table de routage.



```
R2# debug ip rip (selected output)^
RIP: received v1 update from 172.30.2.1 on Serial0/0/0
      172.30.1.0 in 1 hops

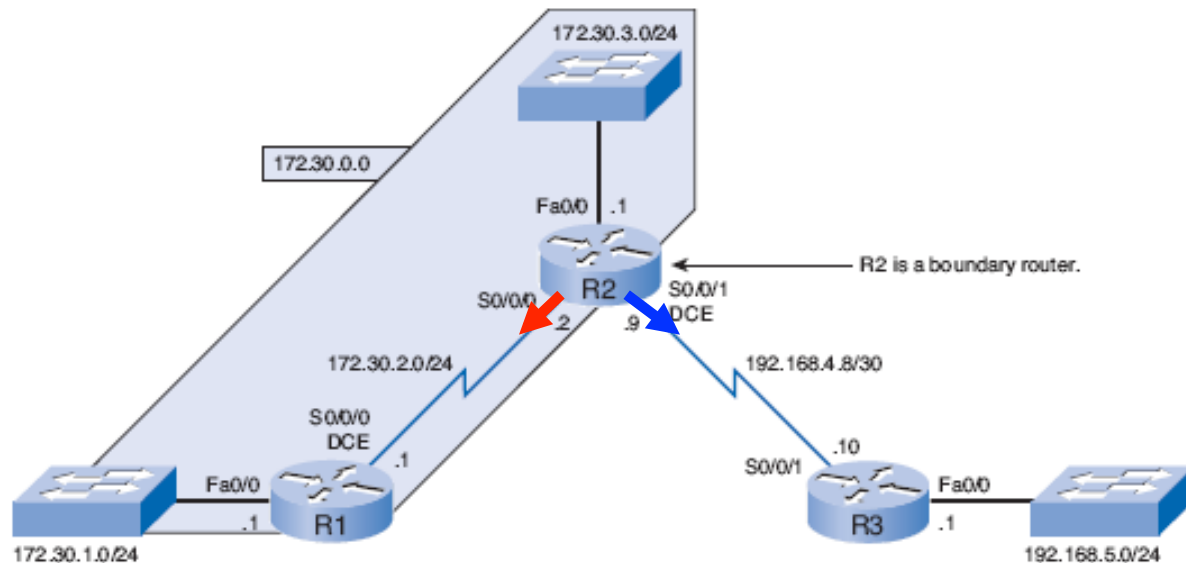
R2# show ip route (selected output)^
      172.30.0.0/24 is subnetted, 3 subnets
R      172.30.1.0 [120/1] via 172.30.2.1, 00:00:18, Serial0/0/0
```



- ▶ Même principe du traitement des updates reçus : Déterminer si on doit ou pas agréger les routes envoyées
  - Quel est le réseau classful de l'interface de sortie ?
  - Quel est le réseau classful du réseau à envoyer en update ?
  - Sont ils dans le même réseau classful ?
    - Oui : Envoie les adresses des sous-réseaux
    - Non : Envoie une adresse agrégée – le réseau classful

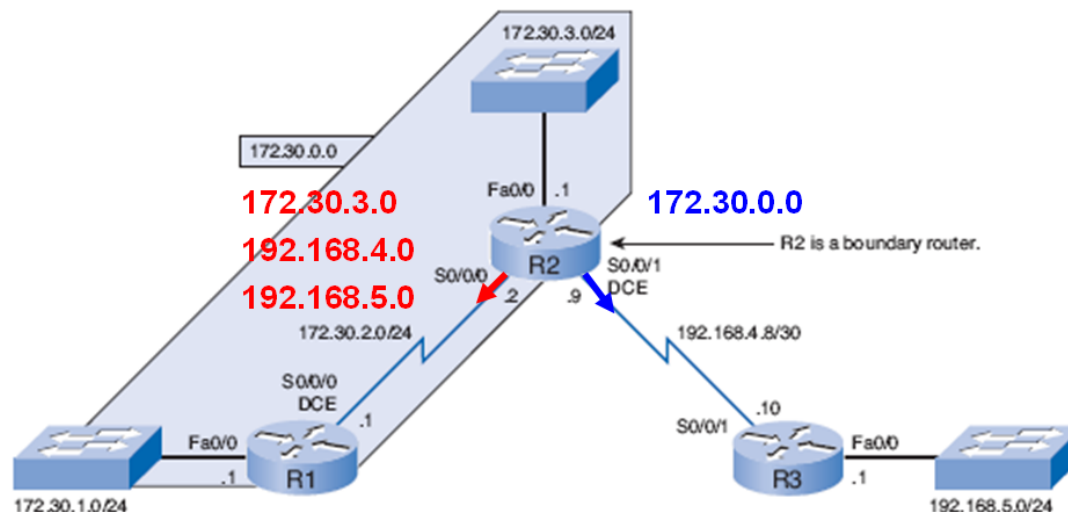
# Envoi d'updates RIP

```
R2# debug ip rip
RIP protocol debugging is on
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (172.30.2.2)
RIP: build update entries
    network 172.30.3.0 metric 1
    network 192.168.4.0 metric 1
    network 192.168.5.0 metric 2
RIP: sending v1 update to 255.255.255.255 via Serial0/0/1 (192.168.4.9)
RIP: build update entries
    network 172.30.0.0 metric 1
```



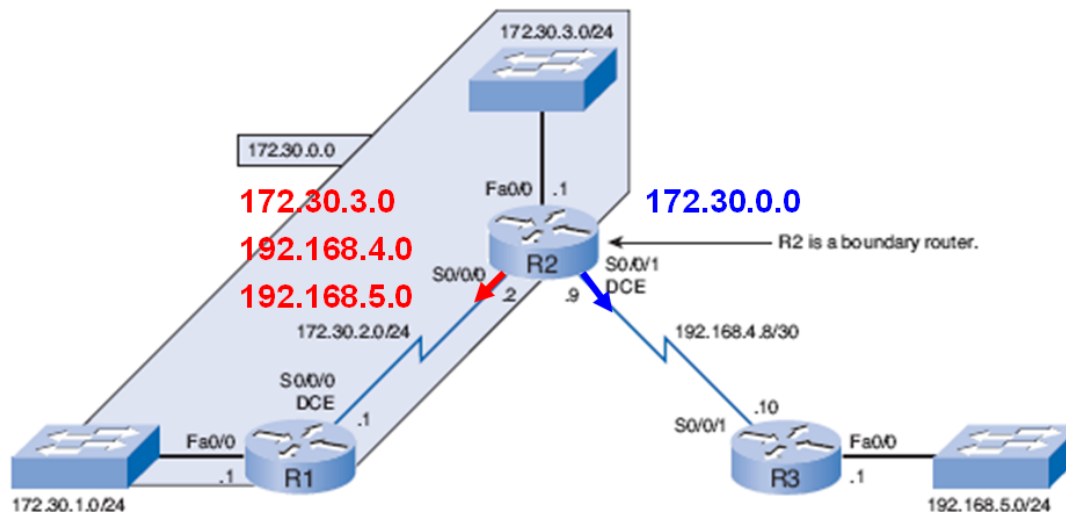
# Envoi d'updates RIP

```
R2# debug ip rip
RIP protocol debugging is on
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (172.30.2.2) ↑
RIP: build update entries
    network 172.30.3.0 metric 1
    network 192.168.4.0 metric 1
    network 192.168.5.0 metric 2
RIP: sending v1 update to 255.255.255.255 via Serial0/0/1 (192.168.4.9) ↑
RIP: build update entries
    network 172.30.0.0 metric 1
```



# Envoi d'updates RIP

```
R2# debug ip rip
RIP protocol debugging is on
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (172.30.2.2) ↑
RIP: build update entries
    network 172.30.3.0 metric 1
    network 192.168.4.0 metric 1
    network 192.168.5.0 metric 2
RIP: sending v1 update to 255.255.255.255 via Serial0/0/1 (192.168.4.9) ↑
RIP: build update entries
    network 172.30.0.0 metric 1
```

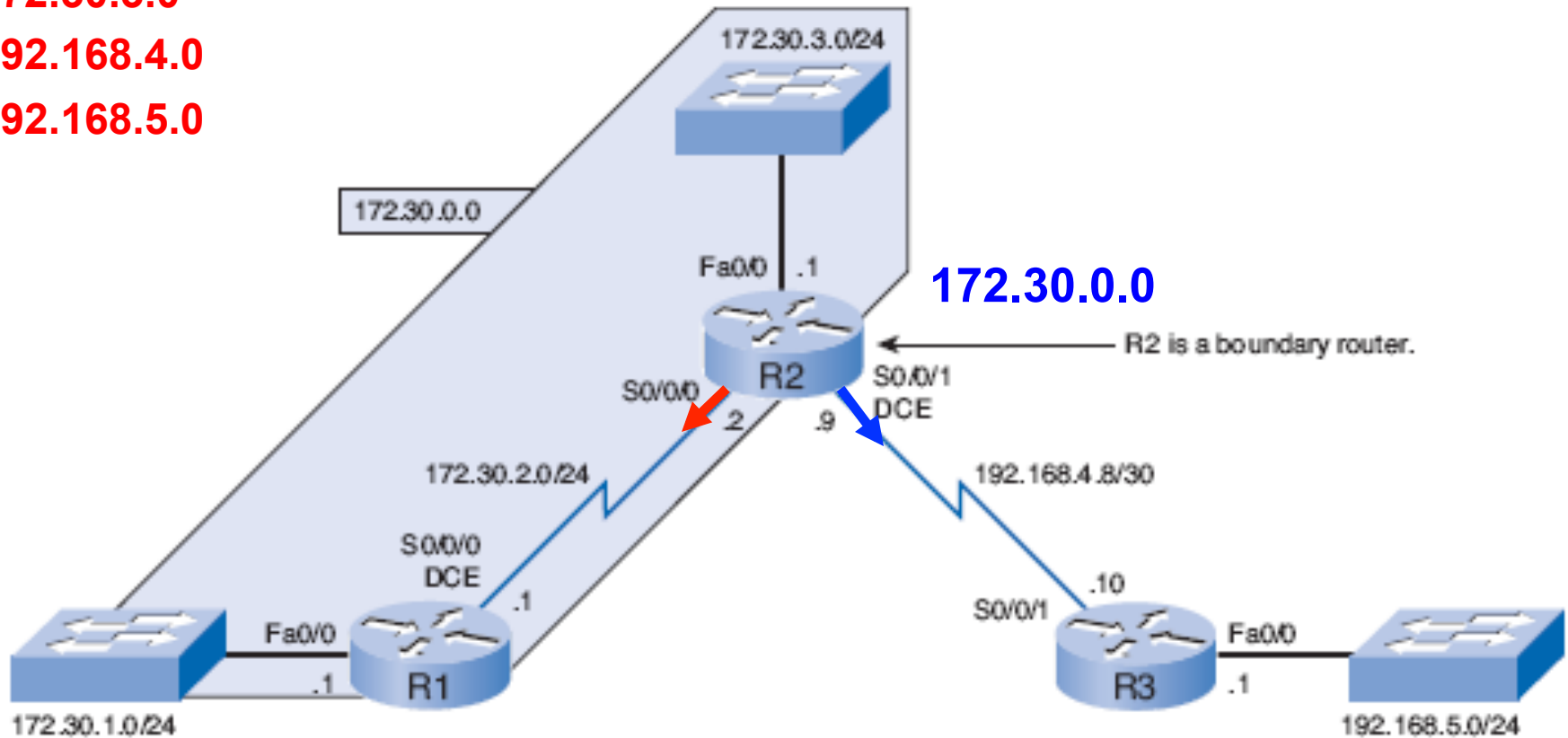


# Envoi d'updates RIP

172.30.3.0

192.168.4.0

192.168.5.0



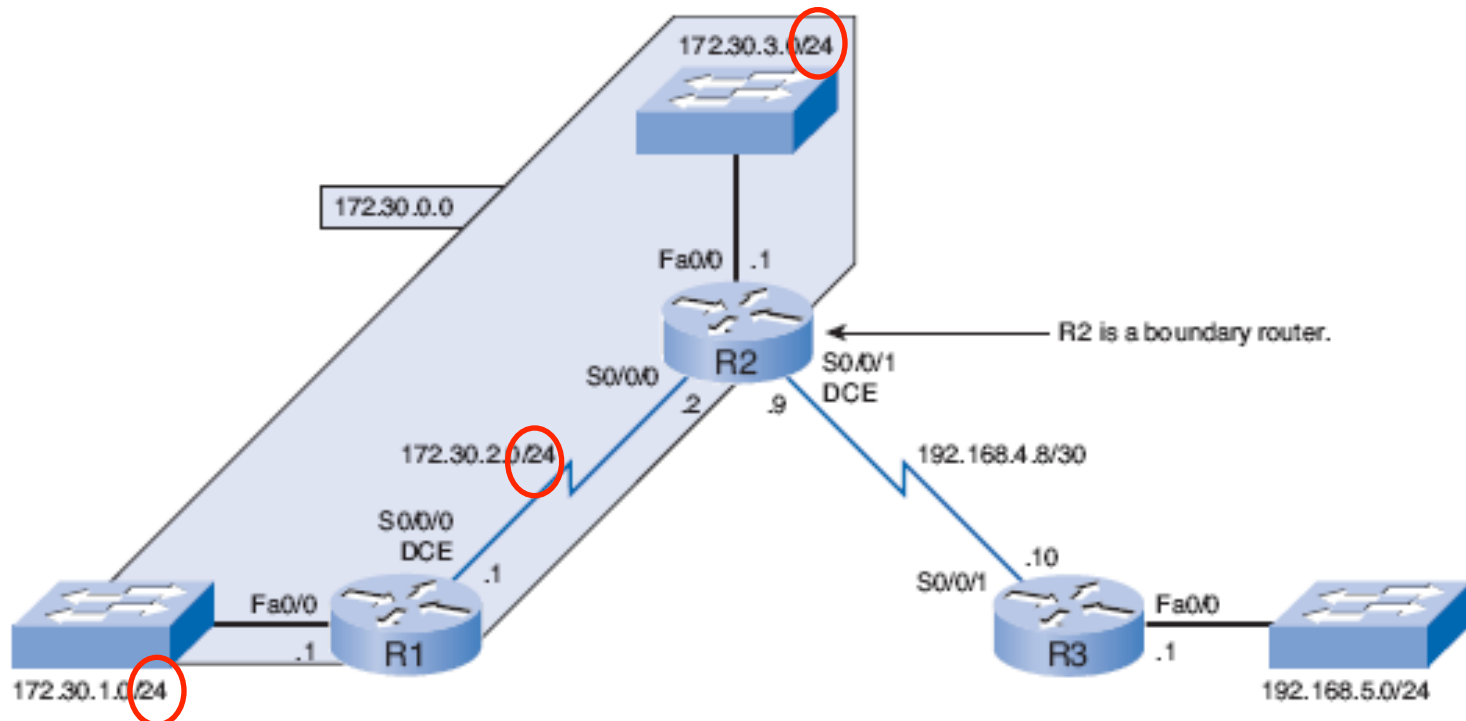
# Vérification des Updates de routage

```
R1# show ip route
<output omitted>
Gateway of last resort is not set
    172.30.0.0/24 is subnetted, 3 subnets
C       172.30.1.0 is directly connected, FastEthernet0/0
C       172.30.2.0 is directly connected, Serial0/0/0
R       172.30.3.0 [120/1] via 172.30.2.2, 00:00:17, Serial0/0/0
R       192.168.4.0/24 [120/1] via 172.30.2.2, 00:00:17, Serial0/0/0
R       192.168.5.0/24 [120/2] via 172.30.2.2, 00:00:17, Serial0/0/0
```

```
R3# show ip route
<output omitted>
Gateway of last resort is not set
R       172.30.0.0/16 [120/1] via 192.168.4.9, 00:00:15, Serial0/0/1
    192.168.4.0/30 is subnetted, 1 subnets
C       192.168.4.8 is directly connected, Serial0/0/1
C       192.168.5.0/24 is directly connected, FastEthernet0/0
```

# Les protocoles classful ne supportent pas le VLSM

- ▶ Les routeurs RIPv1 sont **limités par le masque des réseaux classful**.
- ▶ Nous verrons plus tard que les protocoles classless permettent l'utilisation de masques différents du masque classful (le *variable-length subnet masking* - VLSM)



# Avantages de l'Agrégation Automatique

- ▶ Des messages d'update plus petits sont reçus et envoyés, ce qui réduit l'utilisation de la bande passante entre R2 et R3.
- ▶ R3 a une seule route vers 172.30.0.0/16, peu importe le nombre de sous-réseaux et la façon comme ils sont découpés.
  - Une seule route permet d'accélérer le processus de recherche dans la table de routage de R3.

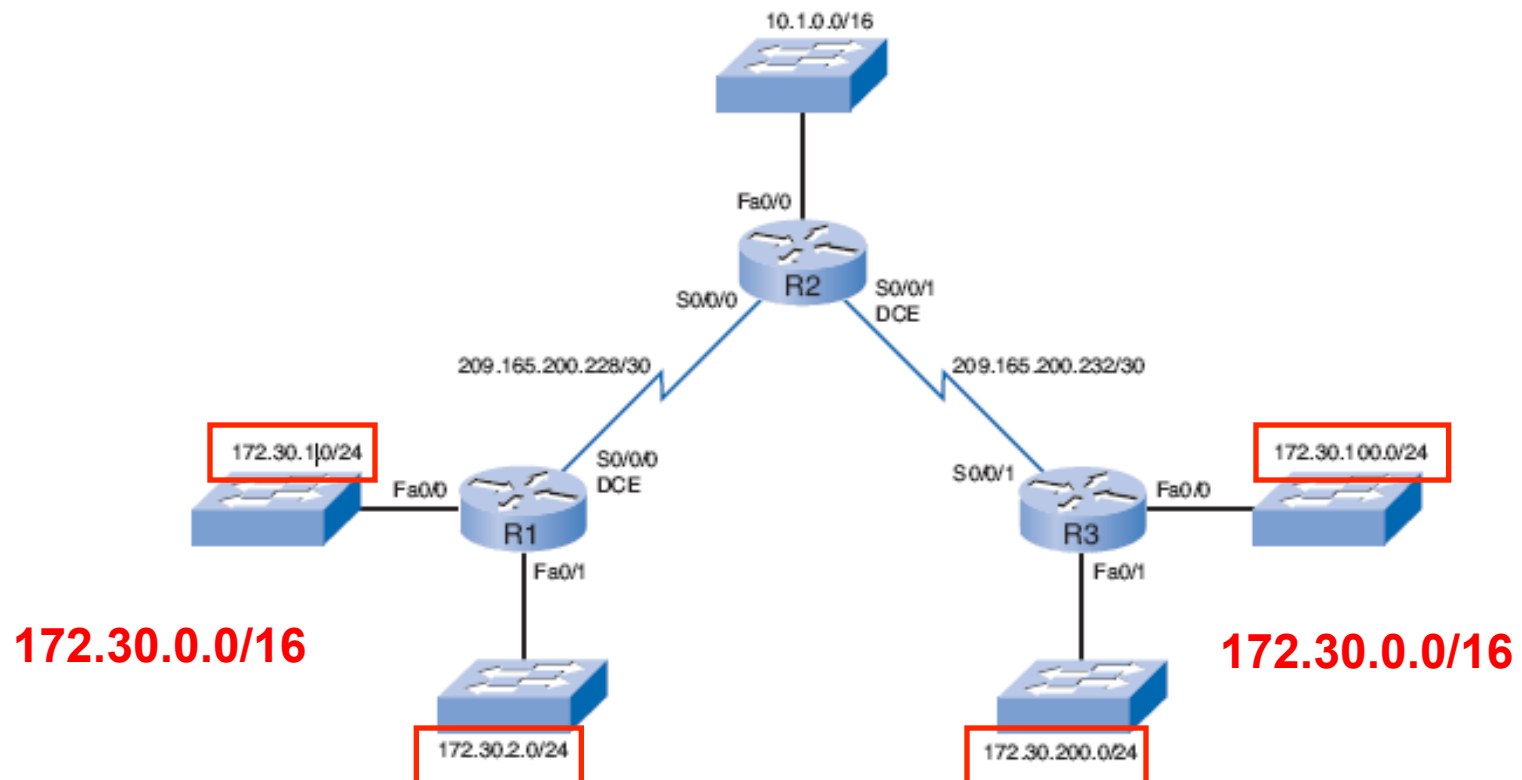
```
R3# show ip route
<output omitted>
Gateway of last resort is not set
R    172.30.0.0/16 [120/1] via 192.168.4.9, 00:00:15, Serial0/0/1
      192.168.4.0/30 is subnetted, 1 subnets
C      192.168.4.8 is directly connected, Serial0/0/1
C      192.168.5.0/24 is directly connected, FastEthernet0/0
```



# Inconvénients de l'Agrégation Automatique

## Scénario B

- ▶ **Les réseaux non-contigus**, deux ou plus sous-réseaux séparés par une autre réseau (d'une autre classe).
- ▶ 172.30.0.0/16 est un réseau non-contigu.



# Les réseaux non-contigus ne convergent pas

```
R1(config)# router rip  
R1(config-router)# network 172.30.0.0  
R1(config-router)# network 209.165.200.0
```

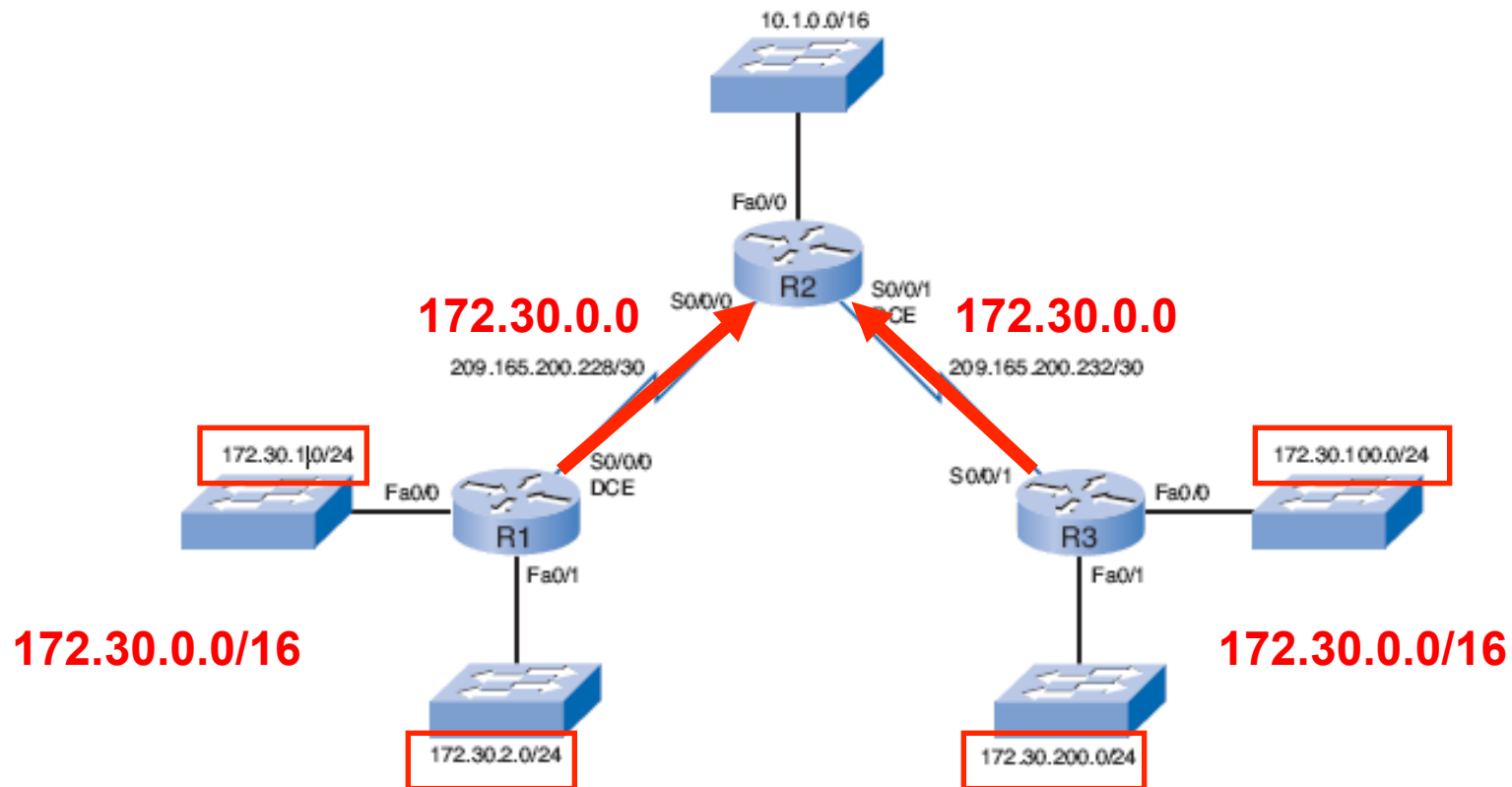
```
R2(config)# router rip  
R2(config-router)# network 10.0.0.0  
R2(config-router)# network 209.165.200.0
```

```
R3(config)# router rip  
R3(config-router)# network 172.30.0.0  
R3(config-router)# network 209.165.200.0
```

- La configuration de RIPv1 est correcte, mais il est incapable de déterminer tous les réseaux dans cette topologie non-contigüe.

# Les réseaux non-contigus ne convergent pas

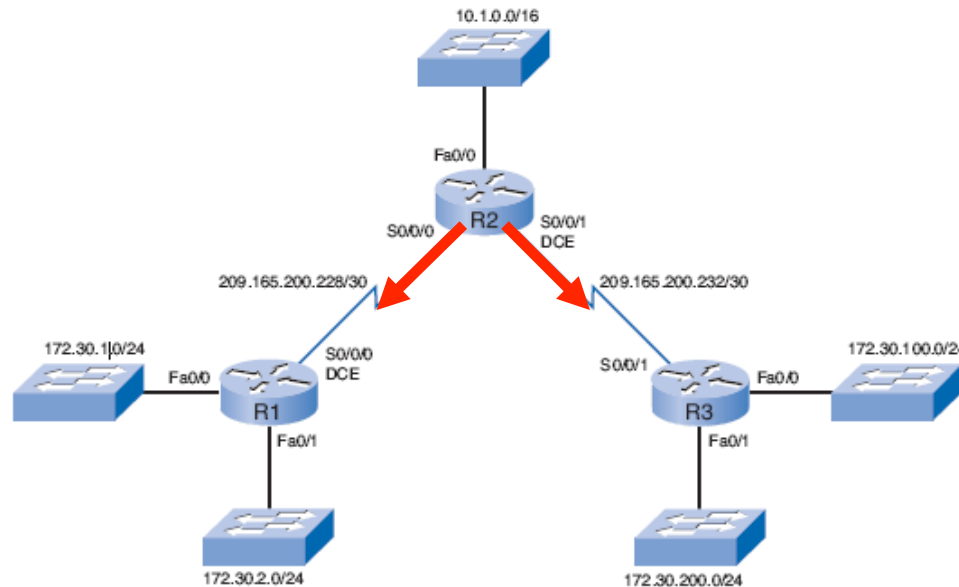
- Les deux routeurs feront l'annonce à R2 du réseau agrégé 172.30.0.0



# Les réseaux non-contigus ne convergent pas

- ▶ R2 a deux routes de même coût pour le réseau 172.30.0.0.
- ▶ R2 fera de l'équilibrage de charge entre les deux liens.
- ▶ Ainsi, R1 aura la moitié des paquets et R3 l'autre moitié, même si le destinataire n'appartient pas à son sous-réseau.

```
R2# show ip route
R    172.30.0.0/16 [120/1] via 209.165.200.234, 00:00:14, Serial0/0/1
      [120/1] via 209.165.200.229, 00:00:19, Serial0/0/0
```



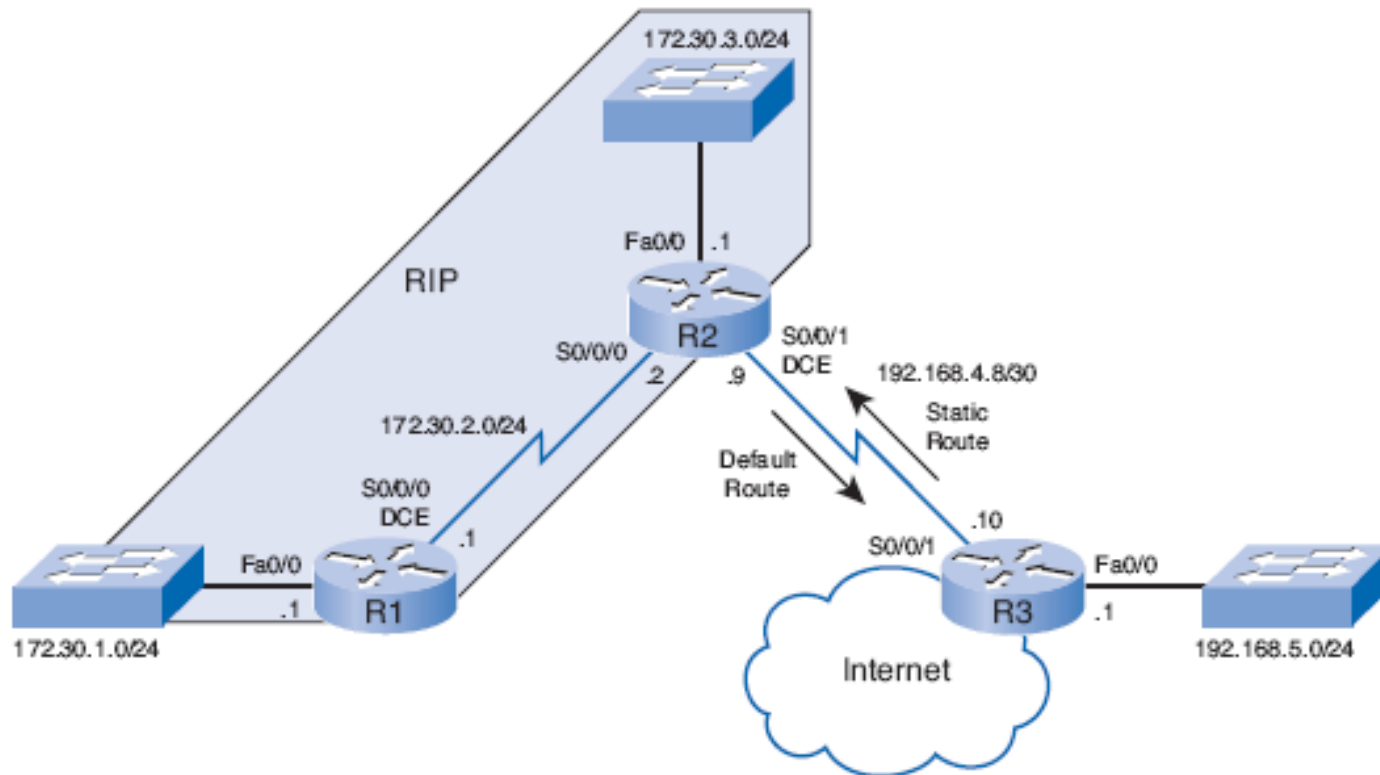
# Route par défaut et RIPv1

---

- Scénario C
- Propagation de la route défaut sur RIPv1

# Topologie modifiée : Scénario C

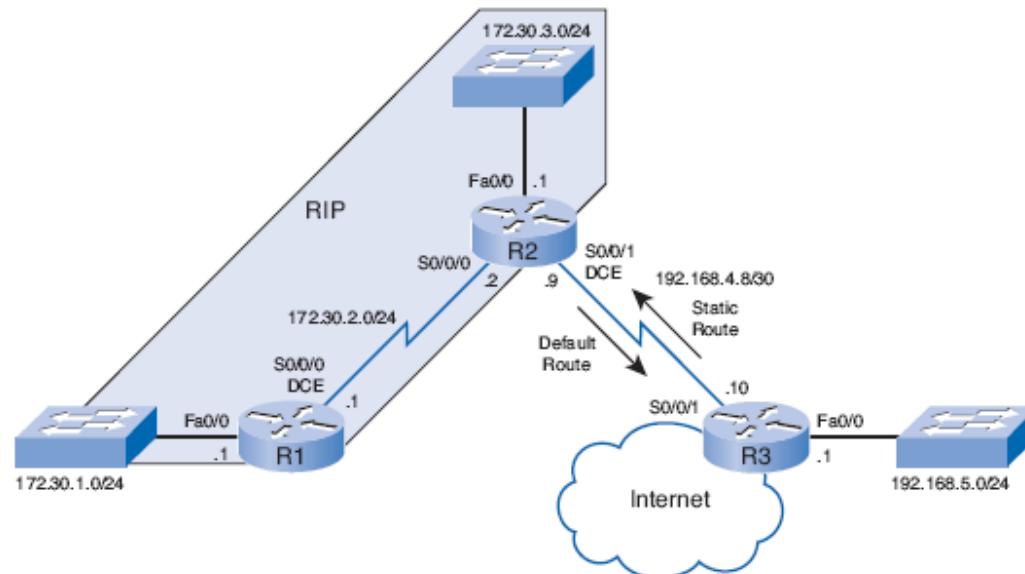
- ▶ Les routes par défaut sont utilisées par les routeurs pour représenter les routes qui ne sont pas présentes sur la table de routage.
- ▶ Une route par défaut est généralement utilisée pour indiquer les réseaux qui ne sont pas administrés localement, comme l'Internet.



# Modification des Configurations de R2 et R3

```
R2(config)# router rip
R2(config-router)# no network 192.168.4.0
R2(config-router)# exit
R2(config)# ip route 0.0.0.0 0.0.0.0 serial 0/0/1

R3(config)# no router rip
R3(config)# ip route 172.30.0.0 255.255.252.0 serial 0/0/1
```



# La table de routage

- ▶ R1 possède des routes pour tous les sous-réseaux de 172.30.0.0/24 mais il jettera les paquets pour les autres réseaux.
- ▶ Aucune route par défaut

```
R1# show ip route
<output omitted>
Gateway of last resort is not set
    172.30.0.0/24 is subnetted, 3 subnets
C       172.30.1.0 is directly connected, FastEthernet0/0
C       172.30.2.0 is directly connected, Serial0/0/0
R       172.30.3.0 [120/1] via 172.30.2.2, 00:00:05, Serial0/0/0
```



# La table de routage

- ▶ R2 a des routes pour les sous-réseaux de 172.30.0.0/16.
- ▶ R2 a une route défaut statique pour les autres réseaux

```
R2# show ip route
```

```
<output omitted>
```

```
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
```

```
172.30.0.0/24 is subnetted, 3 subnets
```

```
R      172.30.1.0 [120/1] via 172.30.2.1, 00:00:03, Serial0/0/0
```

```
C      172.30.2.0 is directly connected, Serial0/0/0
```

```
C      172.30.3.0 is directly connected, FastEthernet0/0
```

```
192.168.4.0/30 is subnetted, 1 subnets
```

```
C      192.168.4.8 is directly connected, Serial0/0/1
```

```
S*    0.0.0.0/0 is directly connected, Serial0/0/1
```

# La table de routage

- ▶ R3 a une route statique pour le réseau 172.30.0.0/16.
- ▶ Peu importe si 172.30.0.0/16 est subnetté, R3 enverra les paquets à R2.

```
R3# show ip route
<output omitted>
Gateway of last resort is not set
    172.30.0.0/22 is subnetted, 1 subnets
S        172.30.0.0 is directly connected, Serial0/0/1
    192.168.4.0/30 is subnetted, 1 subnets
C        192.168.4.8 is directly connected, Serial0/0/1
C        192.168.5.0/24 is directly connected, FastEthernet0/0
```

# Propagation de la Route défaut en RIPv1

- ▶ On peut configurer une route défaut statique sur chaque routeur mais :
  - C'est inefficace
  - Ne réagit pas aux changements de topologie
- ▶ Dans la plupart des protocoles (dont RIP), nous pouvons utiliser la commande **default-information originate** pour spécifier que le routeur est autorisé à propager la route par défaut dans les updates RIP.

```
R1# show ip route
<output omitted>
Gateway of last resort is not set
    172.30.0.0/24 is subnetted, 3 subnets
C       172.30.1.0 is directly connected, FastEthernet0/0
C       172.30.2.0 is directly connected, Serial0/0/0
R       172.30.3.0 [120/1] via 172.30.2.2, 00:00:05, Serial0/0/0
```

# Propagation de la Route défaut en RIPv1

```
R2(config)# router rip
R2(config-router)# default-information originate
R2(config-router)# end
R2# debug ip rip
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0
(172.30.2.2) ^
RIP: build update entries
      subnet 0.0.0.0 metric 1
      subnet 172.30.3.0 metric 1
```

# Propagation de la Route défaut en RIPv1

- ▶ La route statique par défaut dans R2 a été propagée vers R1 dans un update RIP.
- ▶ R1 est maintenant connecté au LAN de R3 et toute destination Internet.

```
R1# show ip route
```

```
<output omitted>
```

```
* - candidate default, U - per-user static route, o - ODR
```

```
Gateway of last resort is 172.30.2.2 to network 0.0.0.0
```

```
172.30.0.0/24 is subnetted, 3 subnets
```

```
C      172.30.2.0 is directly connected, Serial0/0/0
```

```
R      172.30.3.0 [120/1] via 172.30.2.2, 00:00:16, Serial0/0/0
```

```
C      172.30.1.0 is directly connected, FastEthernet0/0
```

```
R*    0.0.0.0/0 [120/1] via 172.30.2.2, 00:00:16, Serial0/0/0
```

# RIPv1 et RIPv2

- ▶ RIPv2 est une extension de RIPv1, pas un nouveau protocole.
  - **Les adresses des next-hop** incluses dans la table de routage
  - Utilisation de **multicast** pour l'envoi des updates
  - Possibilité d'**authentification**
- ▶ Les deux versions de RIP partagent ces caractéristiques et limitations:
  - Utilisation de **hold-down timers** pour éviter les boucles
  - Utilisation de **split horizon** et **split horizon with poison reverse** aussi pour éviter les boucles
  - Utilisation de **triggered updates** lors des modifications pour accélérer la convergence
  - **Maximum hop count de 15 sauts**

