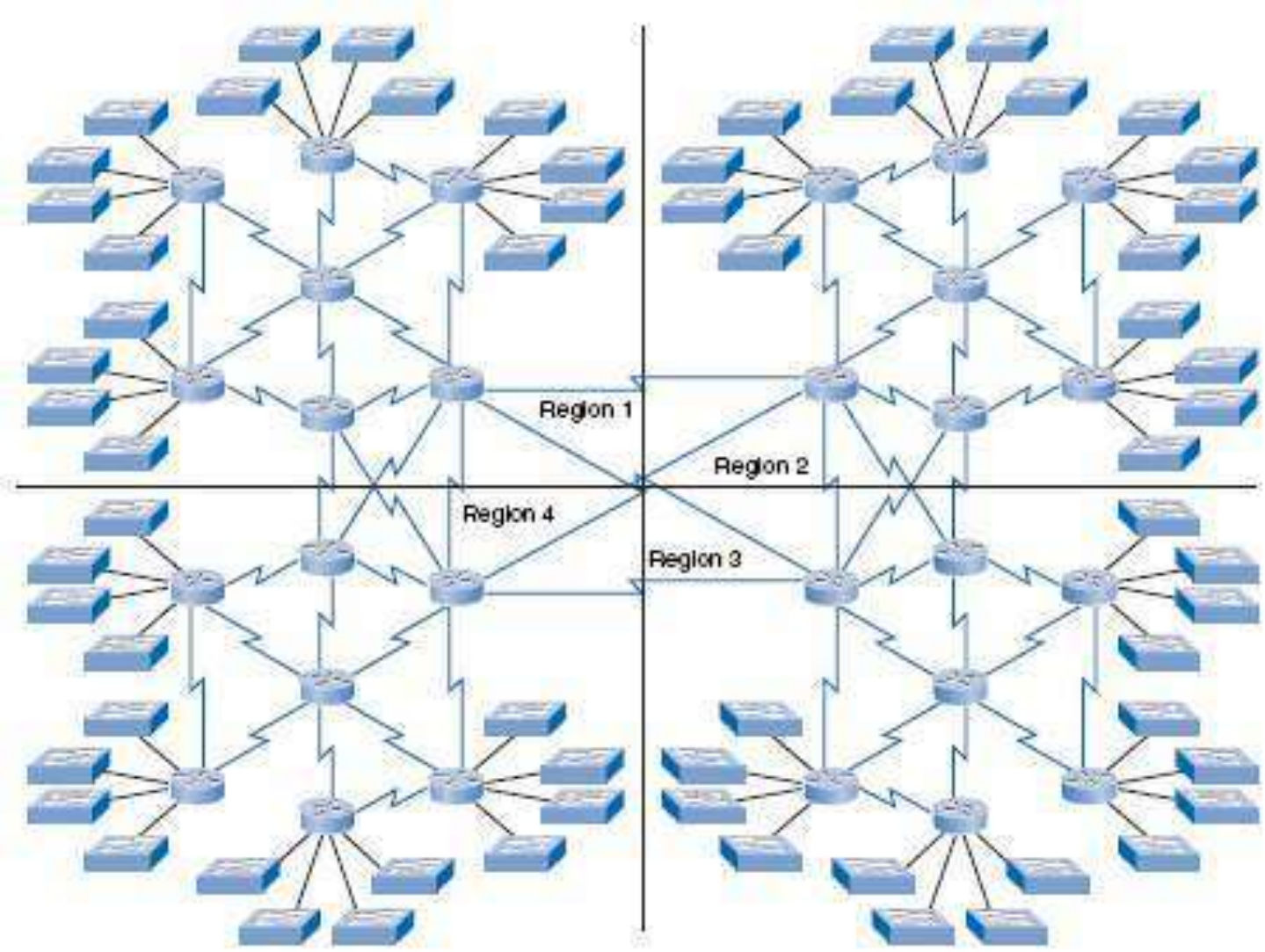
UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Faculté des Sciences Reims

Routage Dynamique

Principes



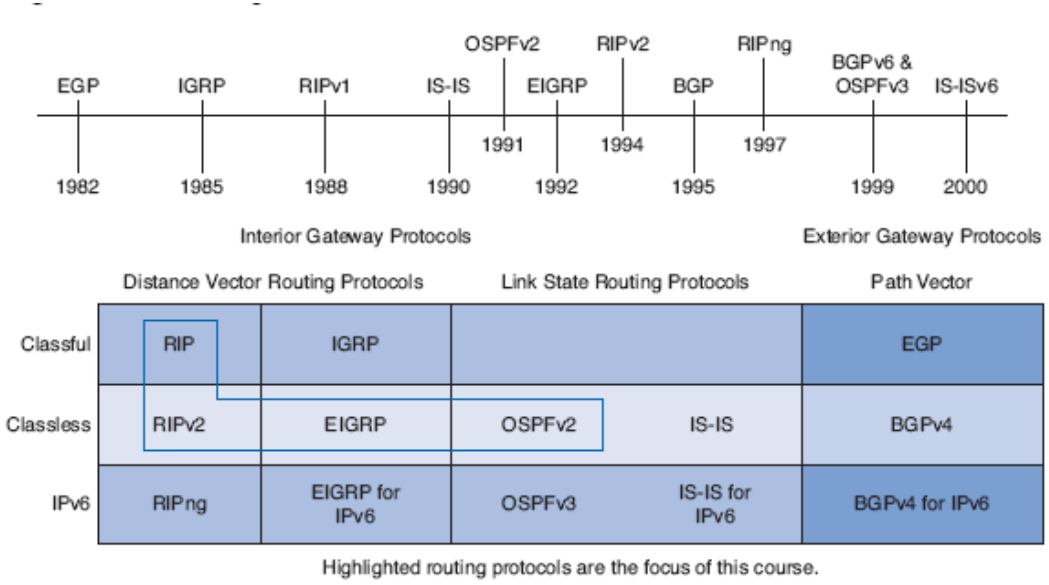
Imaginez-vous gérer les liens statiques dans un réseau pareil

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

CLASSIFICATION DES PROTOCOLES DE ROUTAGE DYNAMIQUE

Classification des protocoles de routage

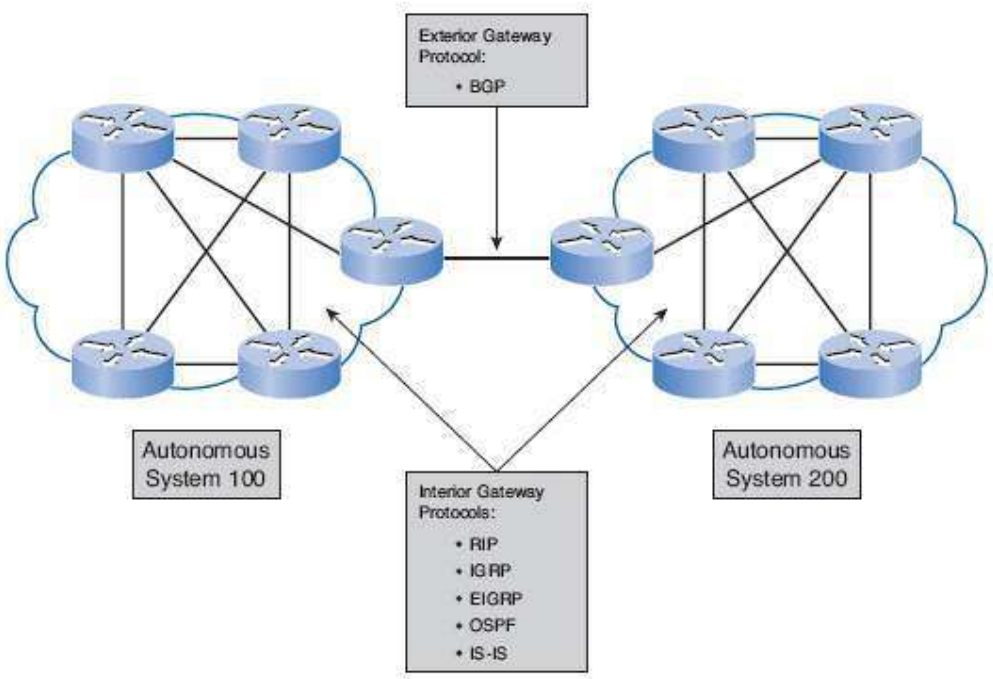
- Les protocoles de routage peuvent être classifiés en :
  - IGP ou EGP
  - Classful ou classless
  - Vitesse de convergence
  - Vecteur de distance ou État des liens



Highlighted routing protocols are the focus of this course.

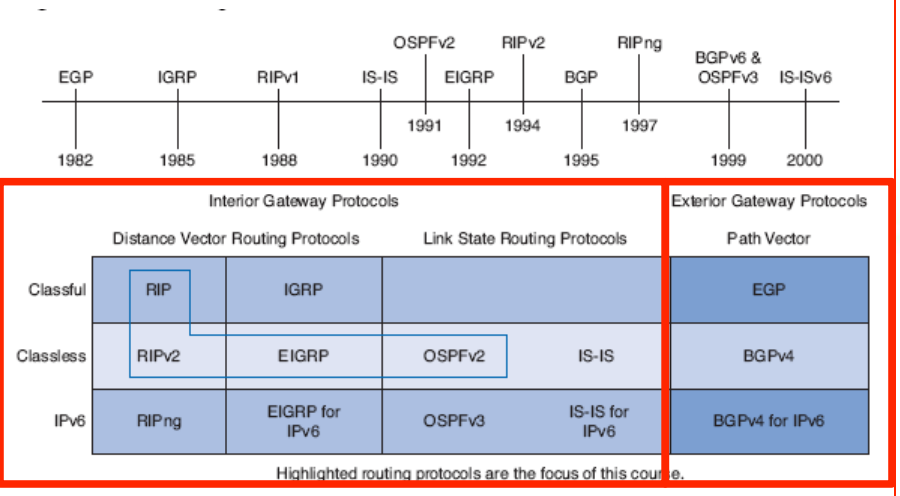
IGP et EGP

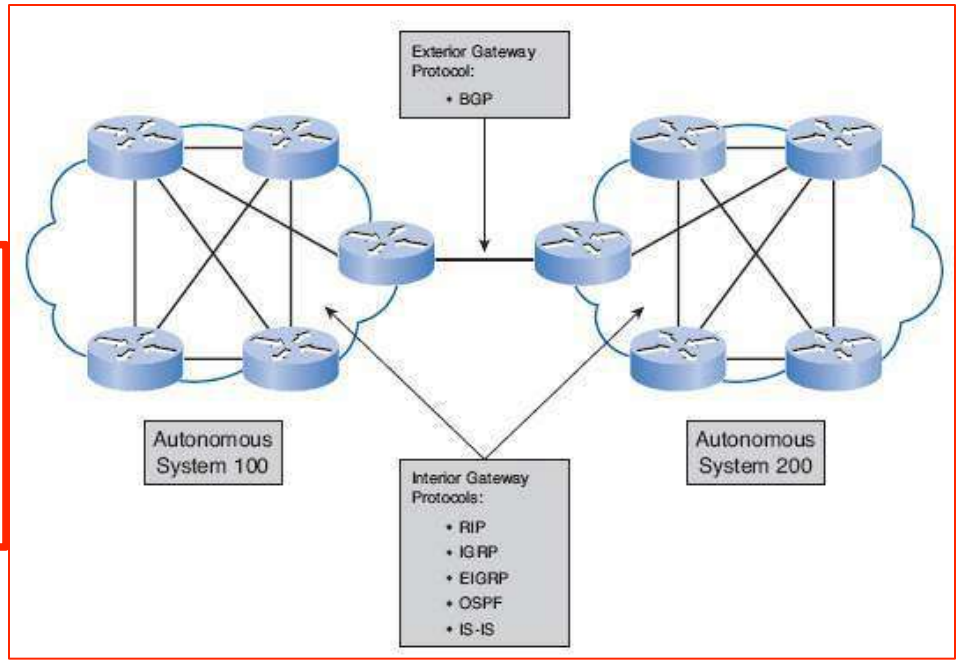
- Un système autonôme (***autonomous system* – AS**), aussi connu sous le nome de ***domaine de routage* –** est l'ensemble de routeurs sous une administration commune
  - Le réseau interne d'une entreprise
  - Le réseau d'un FAI
- Deux types de protocole de routage
  - **Protocoles de routage intérieur**
  - **Protocoles de routage extérieur**



IGP et EGP

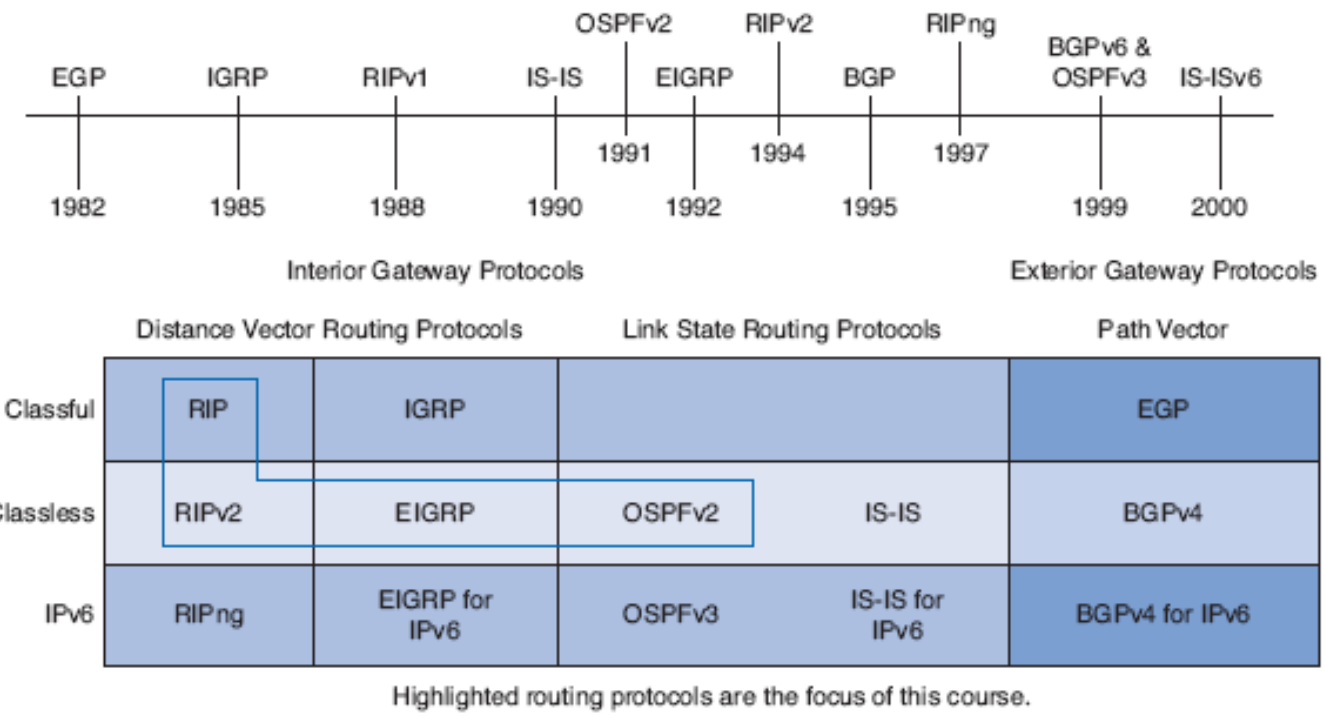
- **Protocole intérieur - Interior gateway protocols (IGP) :**
  - Utilisé pour la propagation des routes à l'intérieur d'un système autonome
- **Protocole extérieur - Exterior gateway protocols (EGP) :**
  - Utilisé pour la propagation des routes entre systèmes autonomes différents





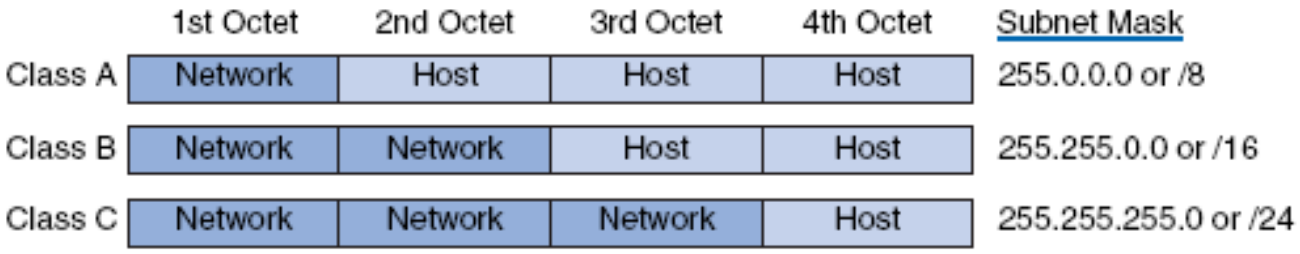
Protocoles Classful et Classless

- Tout protocole de routage peut être classifié soit comme
  - **Protocole de routage Classful** ou
  - **Protocole de routage Classless**
    - Les protocoles de routage pour IPv6 sont toujours classless



Protocoles Classful

- Les protocoles **Classful** n'envoient pas le masque lors des mises à jour de l'information de routage
  - C'est le cas des protocoles plus anciens comme RIP (v1)
  - Conçus à l'époque où les adresses réseau étaient classés
    - Classe A, B, ou C (D et E ne sont pas routés d'habitude)
  - Le protocole de routage n'avait pas besoin d'envoyer le masque
    - Le masque était déterminé selon la valeur du premier octet de l'adresse du réseau



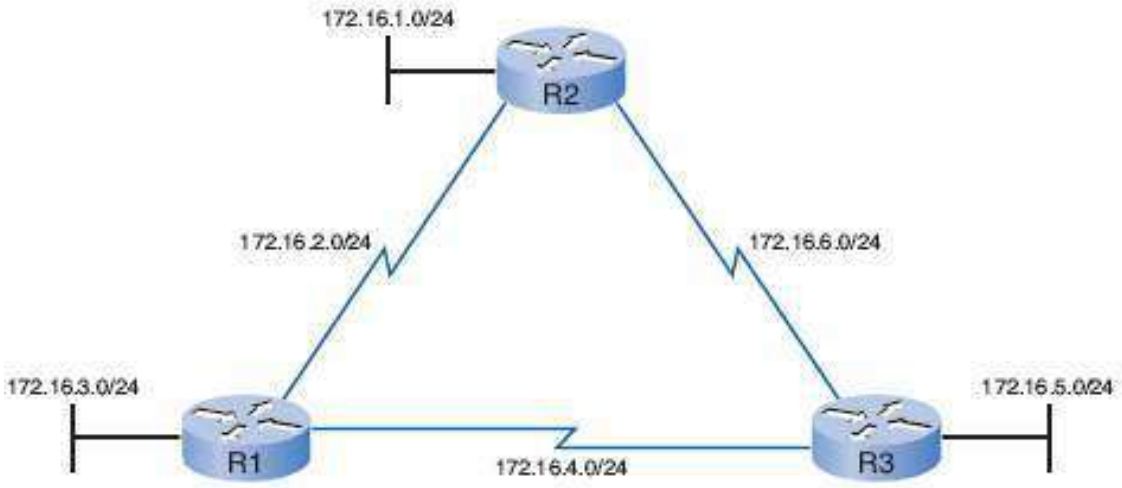
Class	High-Order Bits	Start	End
Class A	0	0.0.0.0	127.255.255.255
Class B	10	128.0.0.0	191.255.255.255
Class C	110	192.0.0.0	223.255.255.255
Multicast	1110	224.0.0.0	239.255.255.255
Experimental	1111	240.0.0.0	255.255.255.255

Protocoles Classful

- Le routage classful n'inclut pas le masque
  - Le masque réseau est obtenu à partir de la classe de l'adresse
- *Tout sous-réseau à l'intérieur d'un même "réseau classful majeur" doit porter le même masque*
- Autres limitations des protocoles classful incluent :
  - Impossibilité de supporter les réseaux non contigus

172.16.0.0/16 réseau classful majeur

Tous sous-réseaux portent le masque /24



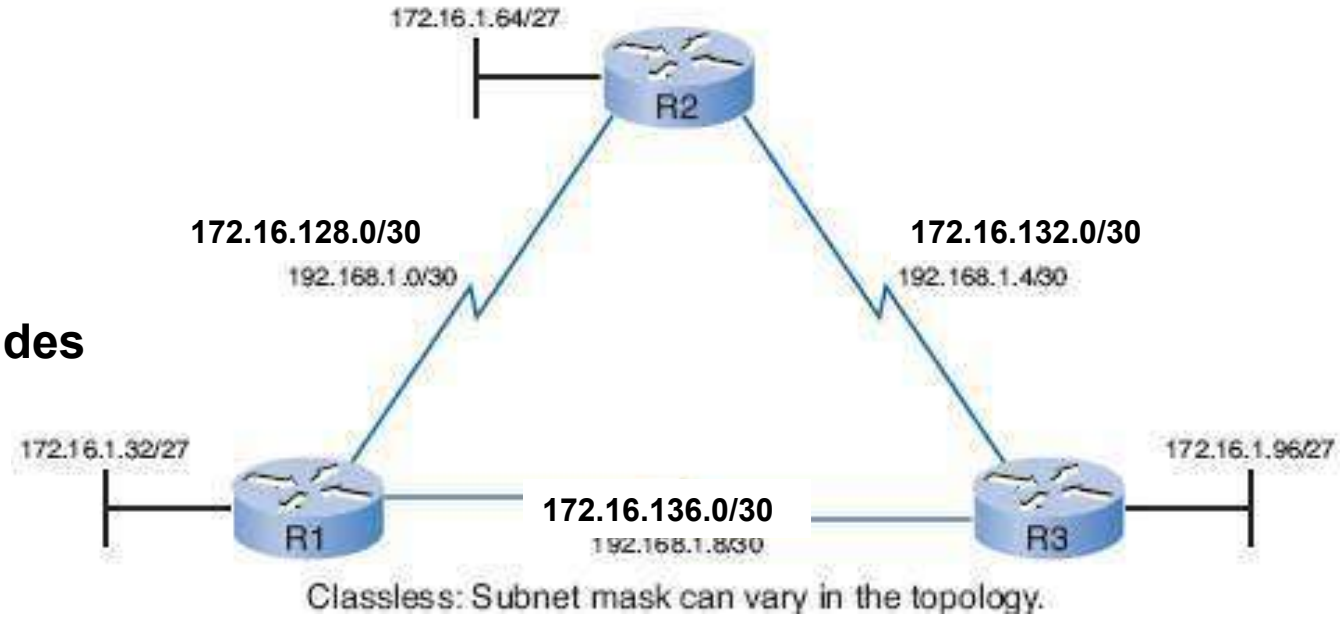
Classful: Subnet mask is the same throughout the topology.

Protocoles Classless

- Les protocoles **Classless** incluent le masque dans les mises à jour
- La plupart des réseaux actuels requièrent des protocoles Classless car ils supportent :
  - VLSM, CIDR et Réseaux non-contigus

172.16.0.0/16 Réseau classful

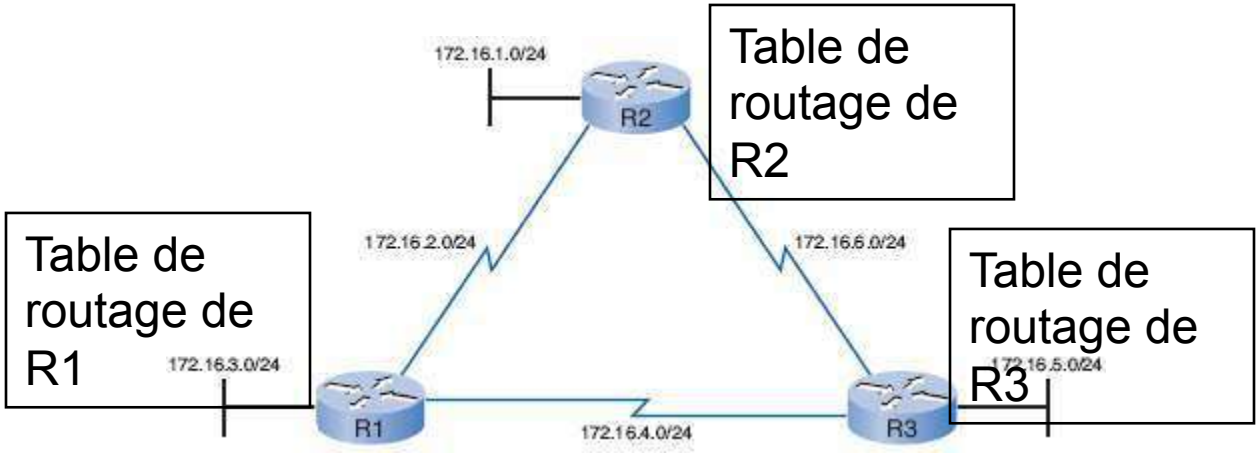
Les sous-réseaux ont des masques /27 et /30



Classless: Subnet mask can vary in the topology.

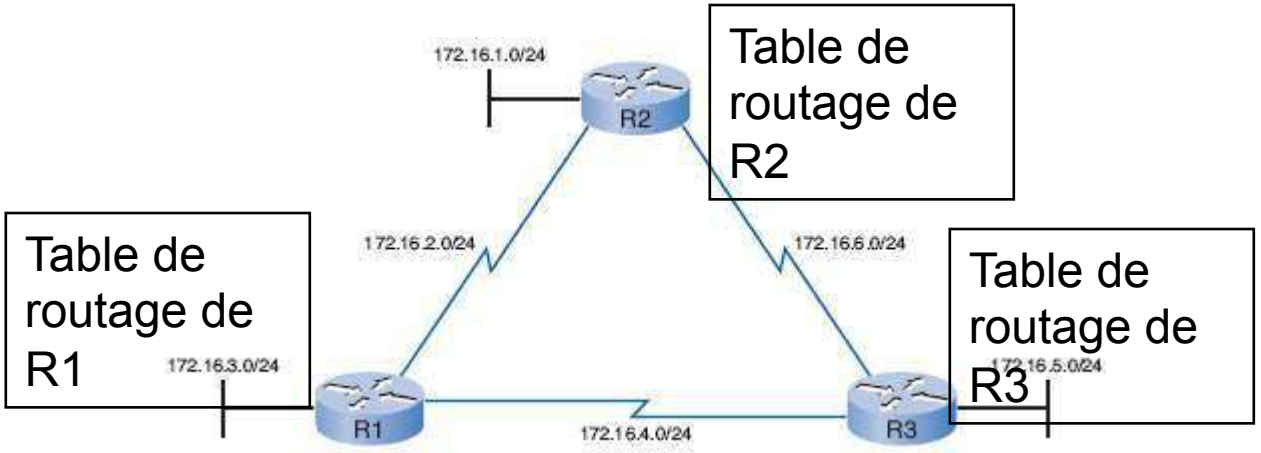
Vitesse de Convergence

- La ***Convergence*** est obtenue lorsque *les tables de routage de tous les routeurs deviennent consistantes*
- Le réseau **a convergé** lorsque tous les routeurs ont une vue complète et précise du réseau



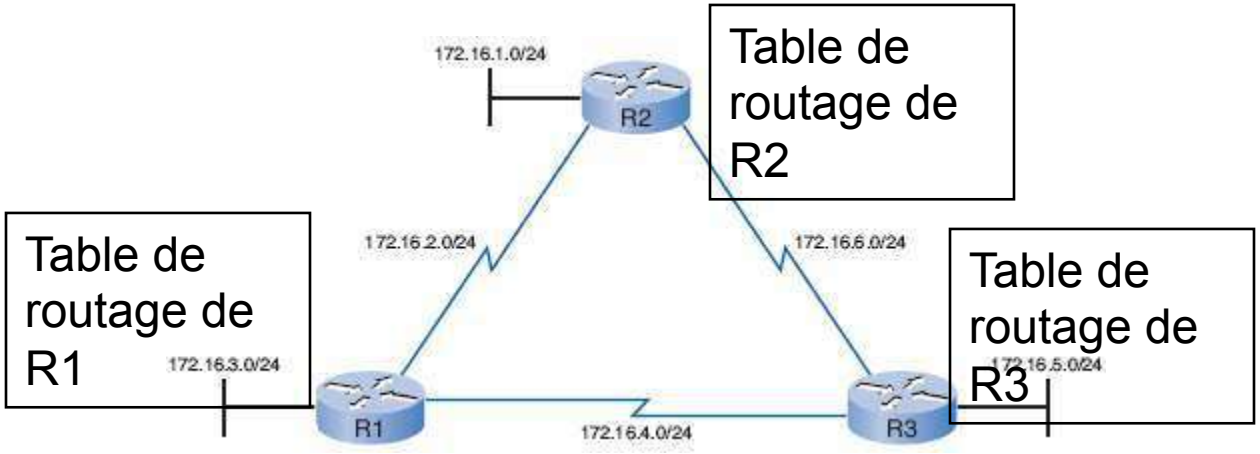
Vitesse de Convergence

- Une caractéristique importante d'un protocole de routage est :
  - La vitesse de convergence lorsqu'un changement est intervenu sur la topologie.
- **Le temps de convergence** concerne le temps pour que les routeurs :
  - Partagent l'information
  - Calculent les meilleurs routes
  - Mettent à jour leurs tables de routage
- Un réseau n'est pas totalement opérationnel tant que le réseau n'a pas convergé ; le temps de convergence devient un facteur critique



Les protocoles dynamiques et la convergence

- De manière générale, le temps de convergence suit :
  - **Lent** : RIP
  - **Rapide** : EIGRP, OSPF, et IS-IS

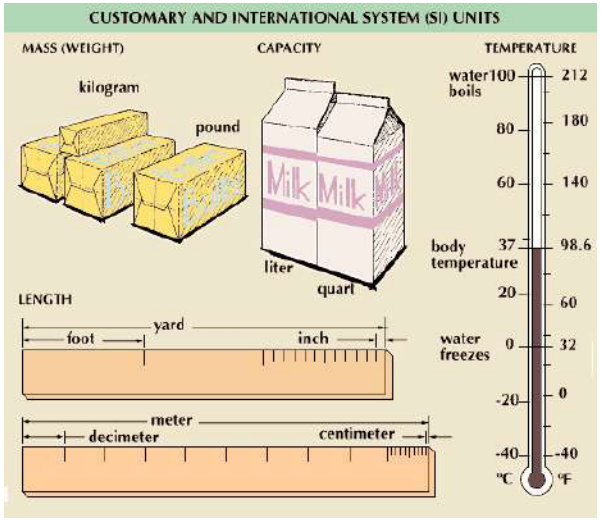


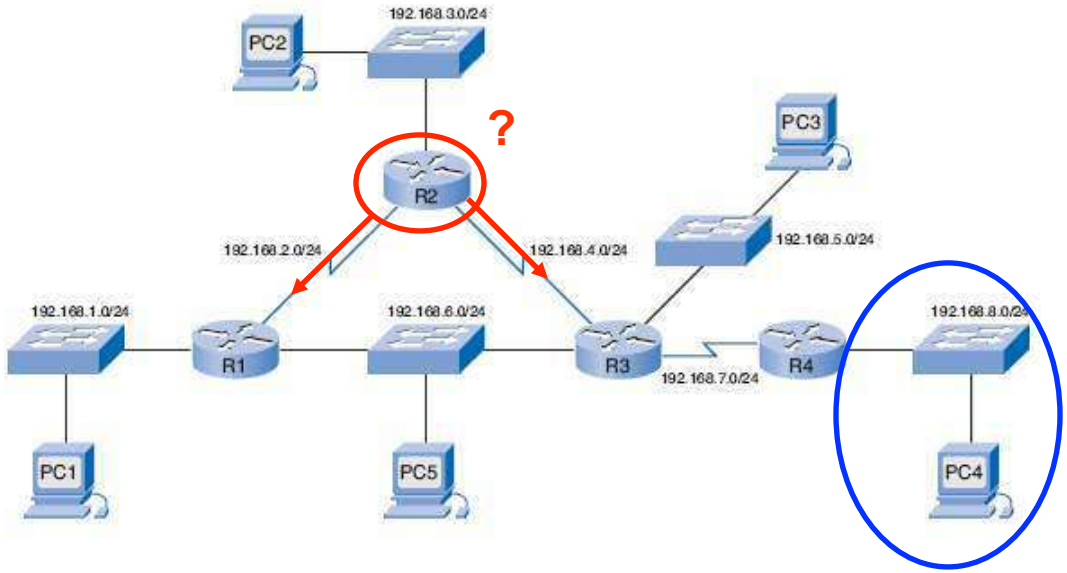
Les Métriques

- Objectifs
- Métriques des différents protos
- Équilibrage de charge

Objectifs des métriques

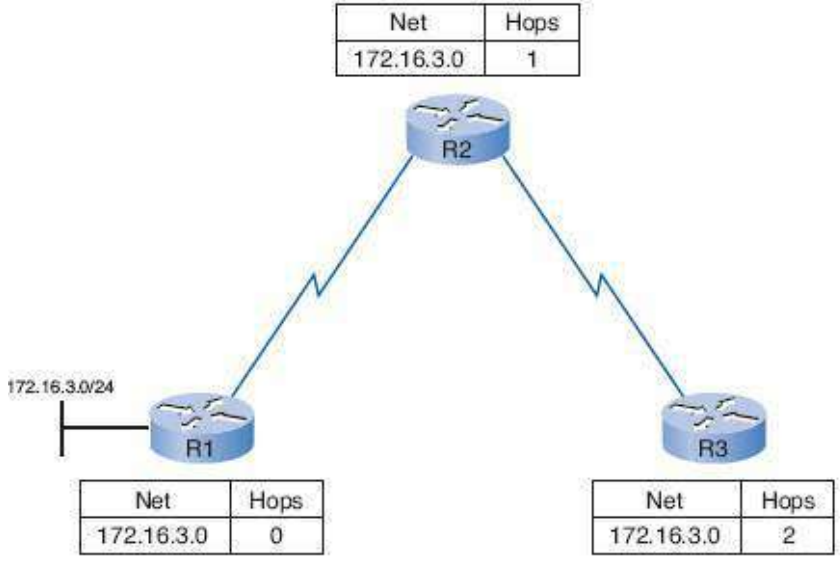
- Les **métriques** sont utilisées pour mesurer ou comparer
  - Déterminer quelle route est la meilleure
  - Attribuer des coûts pour atteindre les réseaux distants
- Les protocoles apprennent plusieurs routes vers une destination
  - Les métriques sont utilisées pour déterminer le chemin préféré





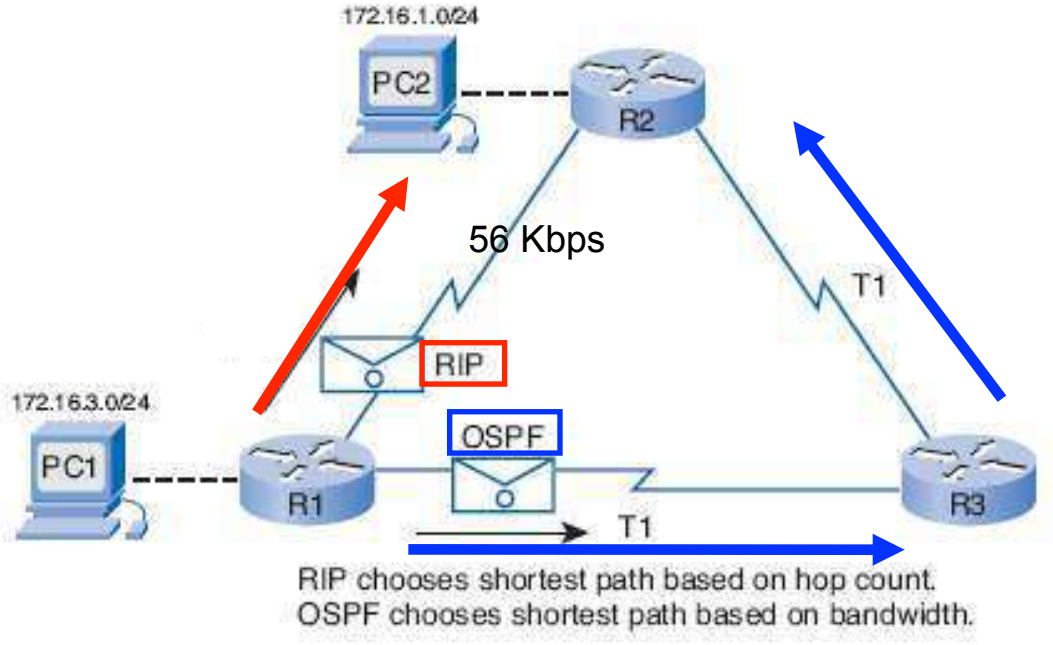
Objectifs des métriques

- Exemples de métriques :
  - **RIP** : Nombre de sauts (Hop count)
  - **EIGRP**: Débit, latence, fiabilité et charge
  - **OSPF** (version Cisco) : Débit



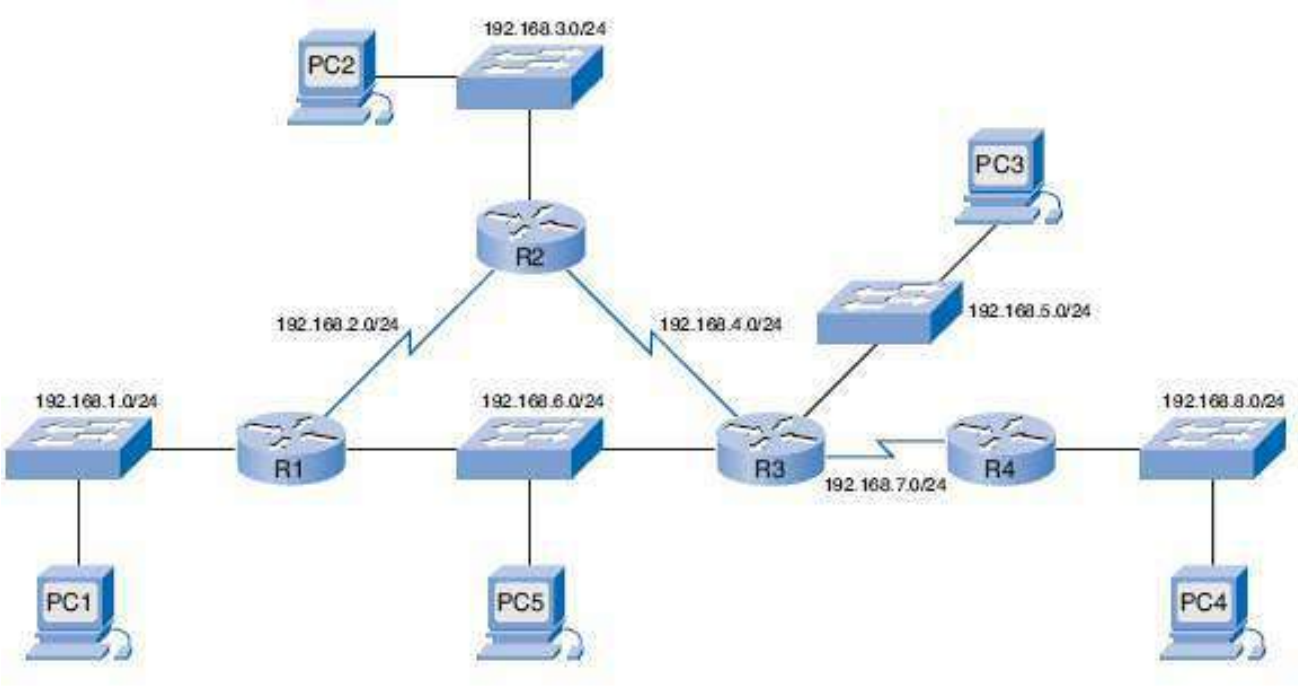
Différences entre les métriques

- **R1** veut atteindre le réseau **172.16.1.0/24**
- **RIP**: Le plus petit nombre de sauts se fait via R2
- **OSPF**: le chemin avec le débit cumulatif plus grand passe par R3
  - Ceci permet l'envoi le plus rapide



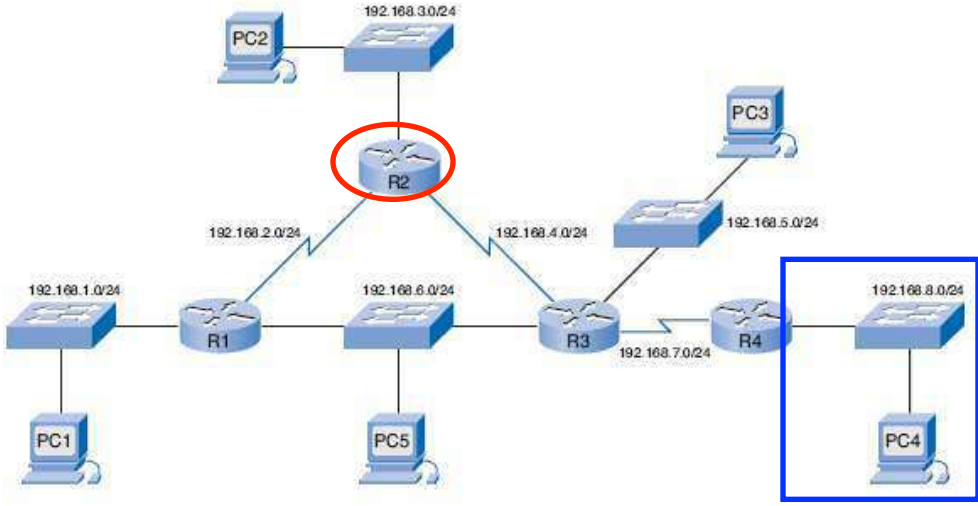
Le champ "métrique" dans la table de routage

- La **table de routage** affiche la métrique pour chaque route statique ou dynamique
  - **Une route statique** a toujours un coût 0 avec RIP
- Les protocoles installent dans la table de routage la route avec la plus petite métrique



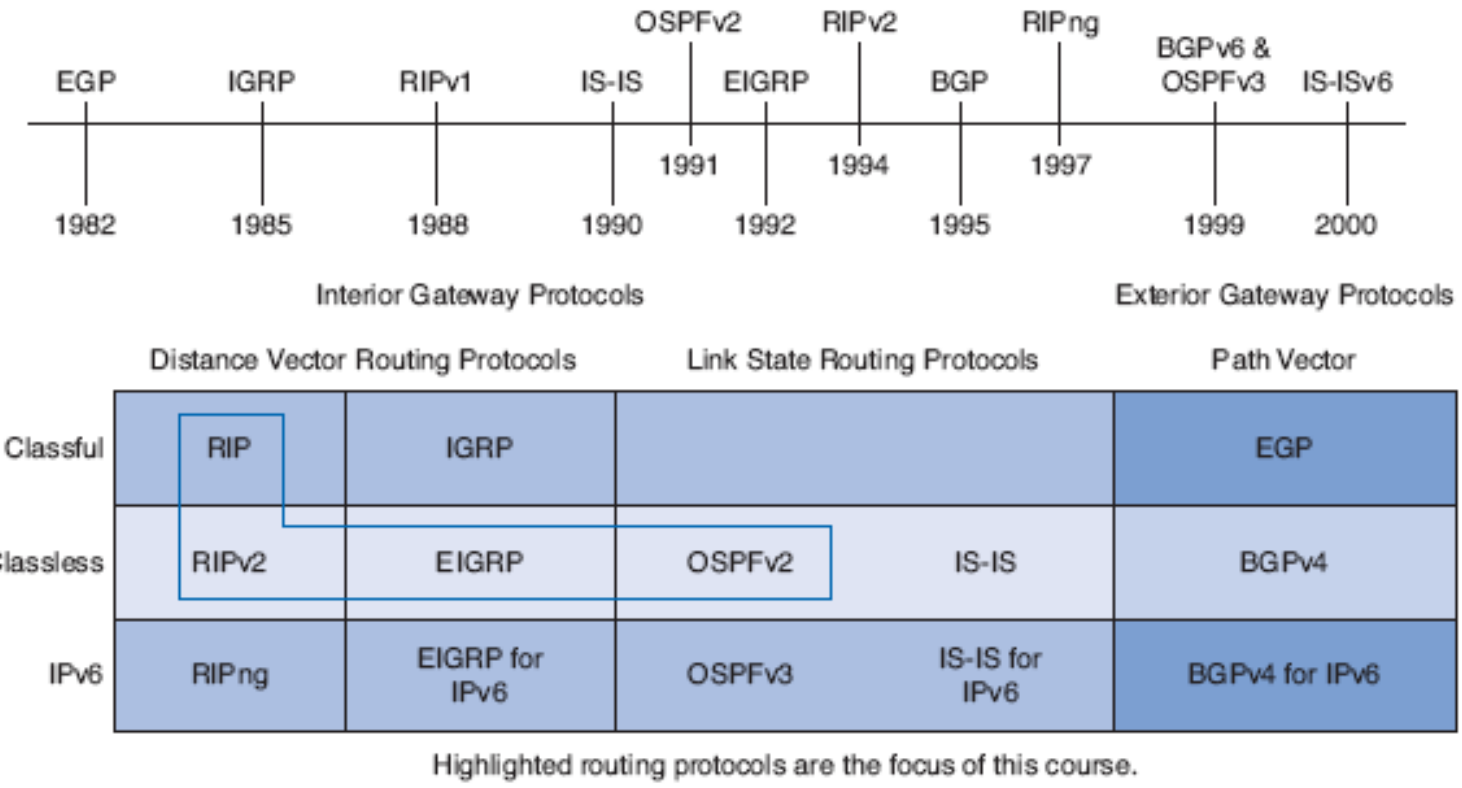
```
R2# show ip route
<output omitted>
Gateway of last resort is not set
R    192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
C    192.168.2.0/24 is directly connected, Serial0/0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/0
C    192.168.4.0/24 is directly connected, Serial0/0/1
R    192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R    192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                    [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R    192.168.7.0/24 [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R    192.168.8.0/24 [120/2] via 192.168.4.1, 00:00:26, Serial0/0/1
```

- Les routeurs tournent RIP
- R2 a une route vers 192.168.8.0/24 avec un coût de **2 sauts**.
- Le **2** indique le coût de la métrique
- **120** est la Distance Administrative



Vecteur de distance ou État des liens

- **Les protocoles intérieurs (IGP)** peuvent être de deux types :
  - **Protocoles à vecteur de distance**
  - **Protocoles à l'état des liens**



Opération d'un protocole à vecteur de distance

- ***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.
- Ces protocoles utilisent souvent l'algorithme **Bellman-Ford** pour la détermination du meilleur chemin

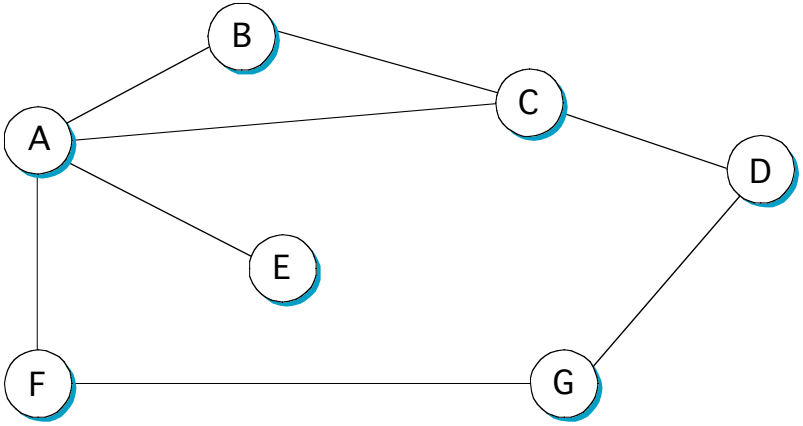
Opération d'un protocole à vecteur de distance

- 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
- Principe similaire à celui des pancartes sur une route

Opération d'un protocole à vecteur de distance

- **Les protocoles à Vecteur de Distance sont indiqués** lorsque :
  - Le réseau est simple et plat et ne requiert pas une structuration hiérarchique
  - L'administrateur n'a pas la connaissance technique pour installer et déboguer un protocole à état des liens.
  - Dans certains types spécifiques de réseaux, telles que le **hub-and-spoke**
  - Lorsque la performance de convergence n'est pas un problème

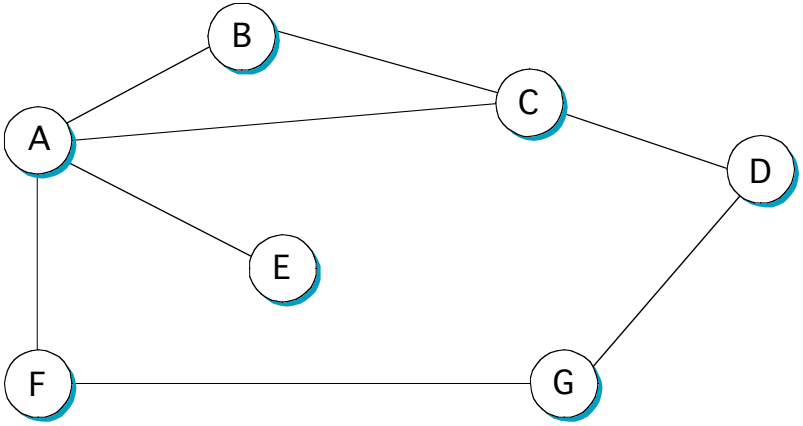
Exemple



- Information sur chaque nœud
- On considère une métrique "saut"

	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

La Table de routage



Avec l'information de départ, la table de routage de A est :

	Coût	Next Hop
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

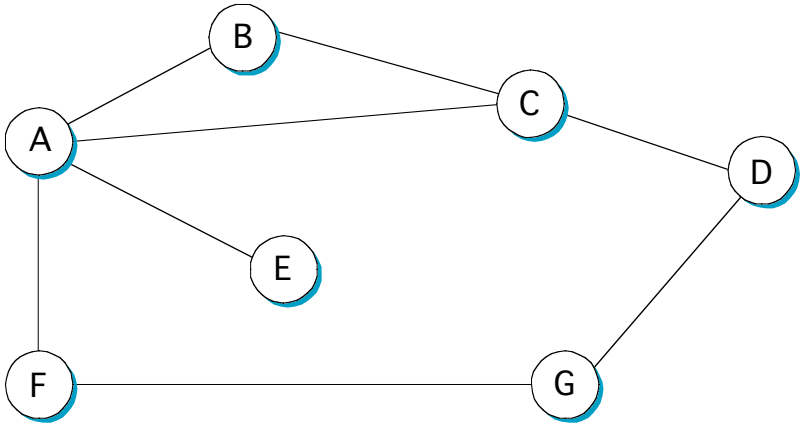
Évolution de la Table de Routage

Chaque nœud envoie à ses voisins le contenu de sa table de routage

Les nouvelles entrées sont mises à jour

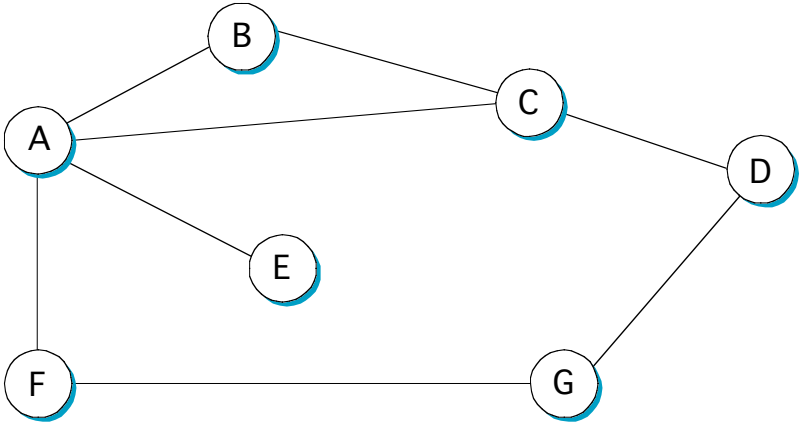
F → A : je connais G à une distance 1

C → A : je connais D à une distance 1



	Coût	Next Hop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Matrice de Distances Finale



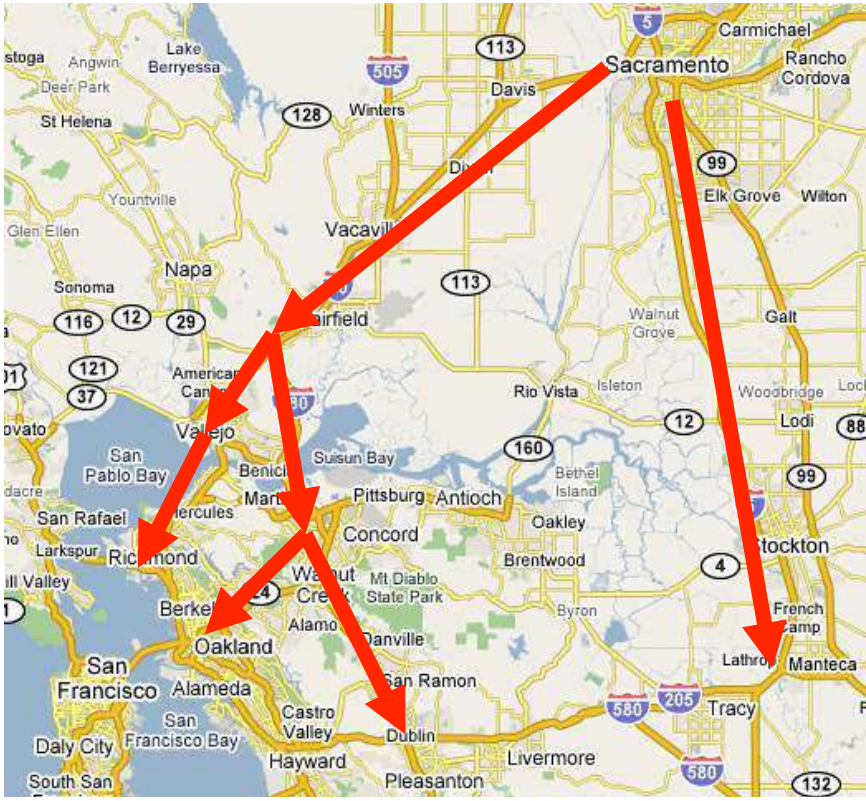
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Attention :

- Plusieurs "vagues" peuvent être nécessaires pour stabiliser la table de routage
- Les envois ne sont pas synchronisés

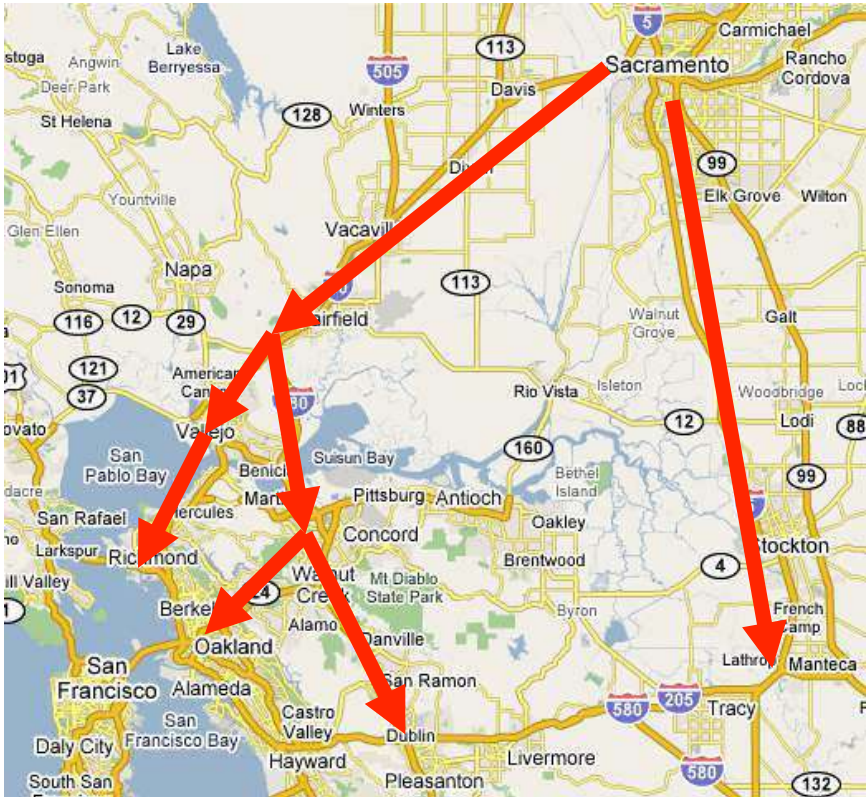
Opération du protocole à état des liens

- Un protocole à état de liens ***(Link-state)*** peut créer une “vue complète” ou topologie, du réseau
- Équivalent à une carte de tout le réseau
- Les protocoles Link-state sont associés à l'algorithme Shortest Path First (SPF) pour l'établissement des routes
- Un routeur ***link-state*** utilise l'information des états des liens pour :
  - Créer une carte topologique
  - Choisir la meilleure route vers toute destination sur la carte



Opération du protocole à état des liens

- **Les protocoles Link-state protocols sont indiqués** pour les situations où
  - La structuration du réseau est hiérarchique, comme dans le cas des grands réseaux
  - L'administrateur a une bonne connaissance du protocole link-state installé
  - Une convergence rapide est cruciale
- On verra des exemples plus détaillés un autre jour



Protocole de Routage ou Protocole Routé ?

- Nous faisons aussi une classification (plus générale) des protocoles en deux types :
  - **Protocoles routés**
    - Tout protocole qui n'est pas capable (tout seul) d'atteindre une destination distante
    - Ex : TCP, HTTP, IP
  - **Protocoles de routage**
    - Protocoles auxiliaires qui permettent le routage des protocoles routés
      - Diffusion de l'information, construction des tables de routage, meilleur chemin
    - Ex : RIP, OSPF, EIGRP, IS-IS, BGP
- Certains protocoles de routage utilisent les services de protocoles routés pour l'établissement de connexions sur le segment
  - Ex : RIP et OSPF utilisent UDP/IP, BGP utilise TCP/IP
- D'autres protocoles de routage utilisent leurs propres services de transport, indépendants des protocoles routés
  - Ex : EIGRP utilise RTP, IS-IS utilise CLNP

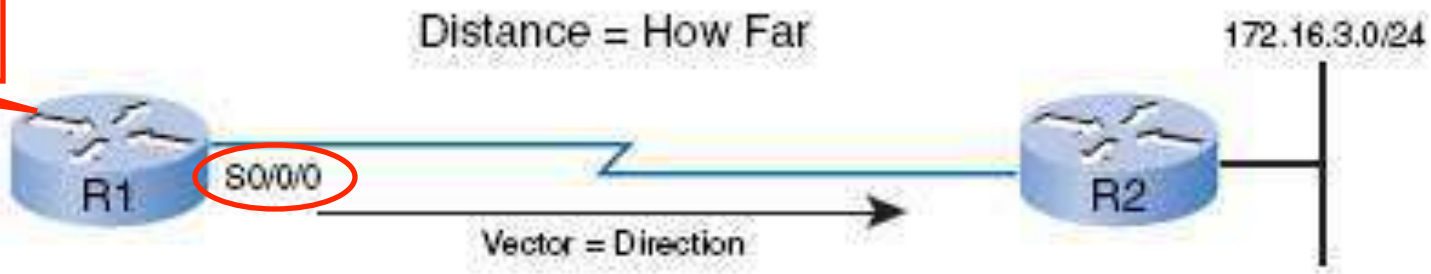
Protocoles à Vecteur de Distances - Précisions

Vecteur de distances : rappel

- ***Vecteur de distances***
  - Les routes sont annoncées comme ***vecteurs de distance et direction***
- Routes annoncées aux voisins directement connectés

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)

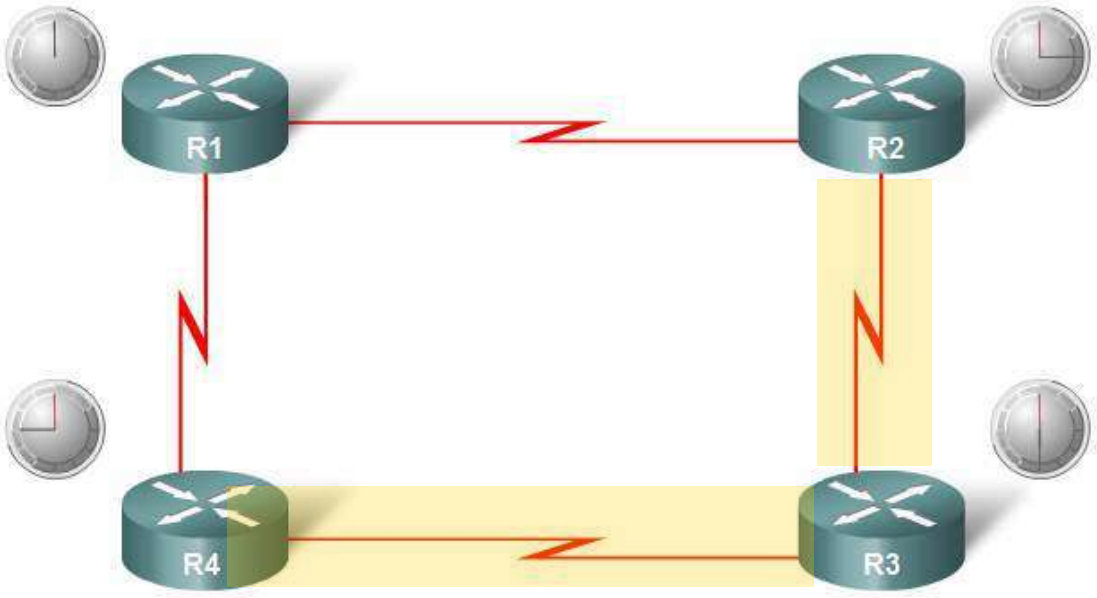
Quelle est la **Direction** ?

S0/0/0

R1 connaît la carte topologique du réseau ?

Non, seulement la distance et la direction !

Opération des protocoles à vecteur de distances



Voisin de R1

Voisin de R1

R1 ne connaît pas R3 et son réseau

Voisins ?

Les voisins sont les routeurs qui :

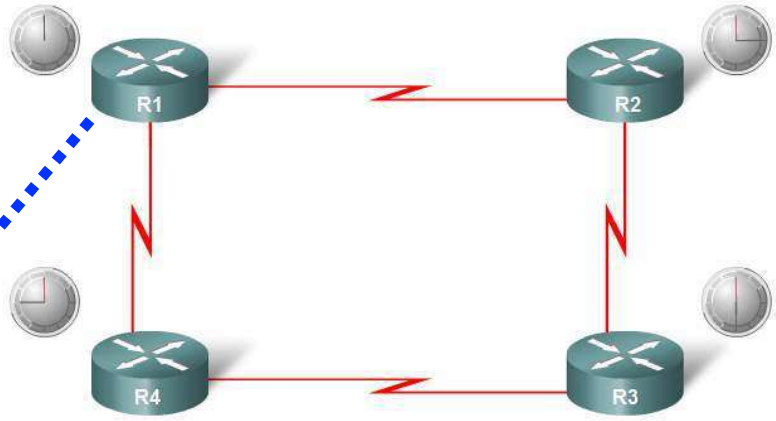
- Partagent un lien
- Utilisent le même protocole de routage

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

- Les adresses de ses interfaces
- Les adresses de ses voisins

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=0, 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)), π (u) = v
→ Step 1

Step 3:judgement

i ← i +1

If i < n, then Q ← 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".

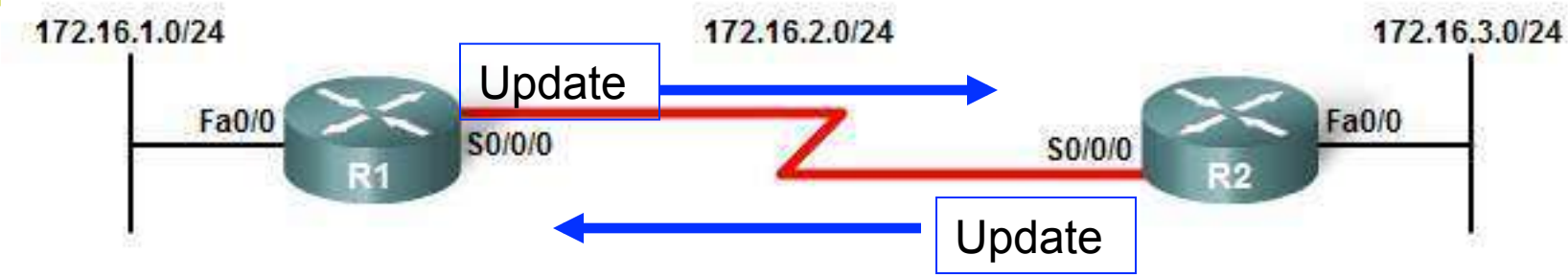
* Triangle inequality

Let X be linear space,

$\| u+v \| \leqq \| u \| + \| v \|$ 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

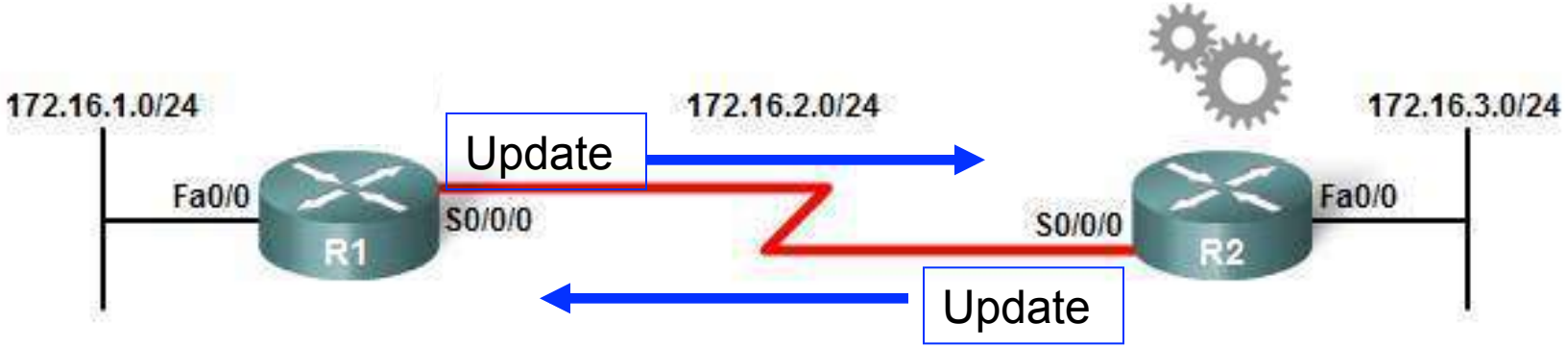


Network	Interface	Hop
172.16.1.0/24	Fa0/0	0
172.16.2.0/24	S0/0/0	0

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



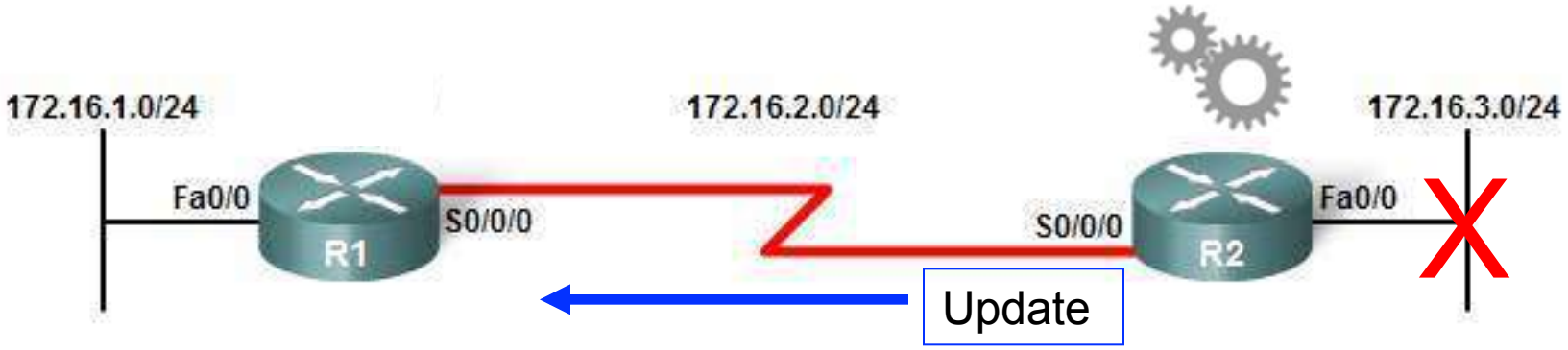
Network	Interface	Hop
172.16.1.0/24	Fa0/0	0
172.16.2.0/24	S0/0/0	0

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



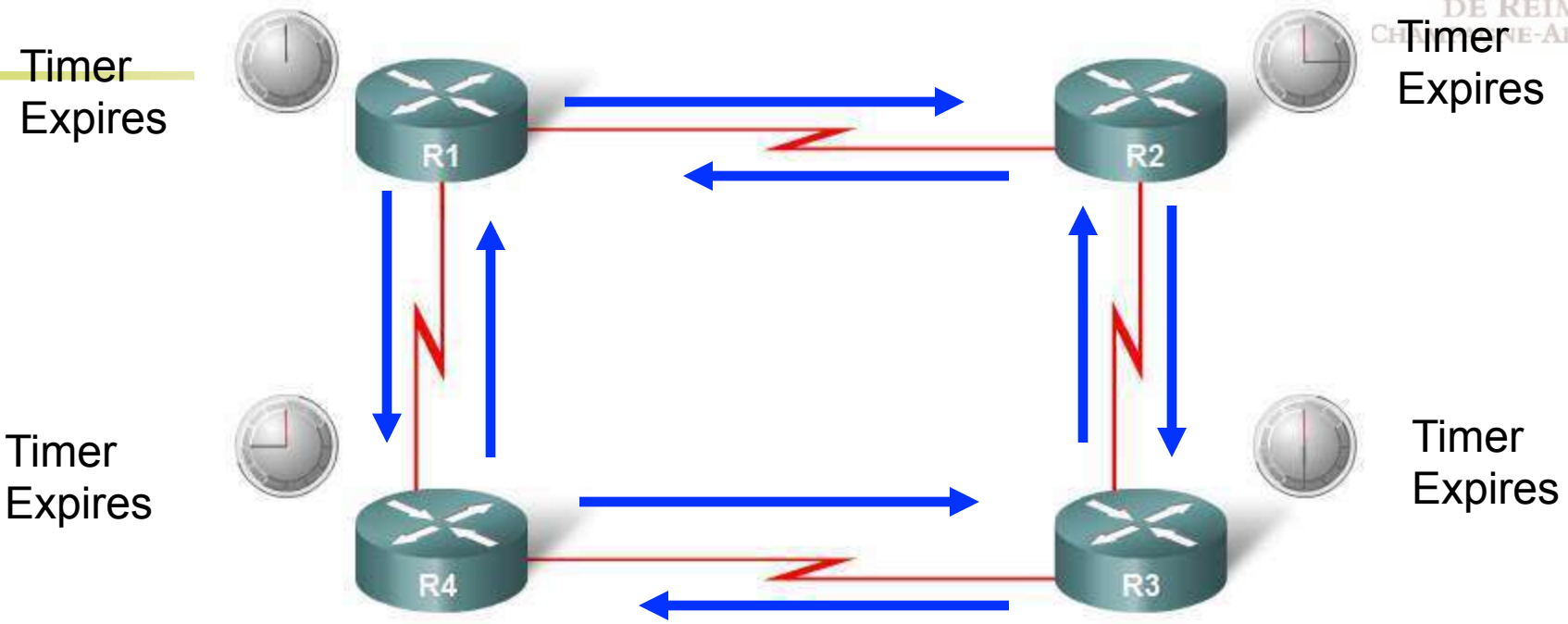
Network	Interface	Hop
172.16.1.0/24	Fa0/0	0
172.16.2.0/24	S0/0/0	0
~~172.16.3.0/24~~	~~S0/0/0~~	~~1~~

Network	Interface	Hop
172.16.2.0/24	S0/0/0	0
~~172.16.3.0/24~~	~~Fa0/0~~	~~0~~
172.16.1.0/24	S0/0/0	1

Dans chaque routeur, l'algorithme :

- Détecte et réagit aux changements de topologie

Opération des protocoles à vecteur de distances



Mises à jour régulières

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

Intervalle de 30 seconds pour RIP

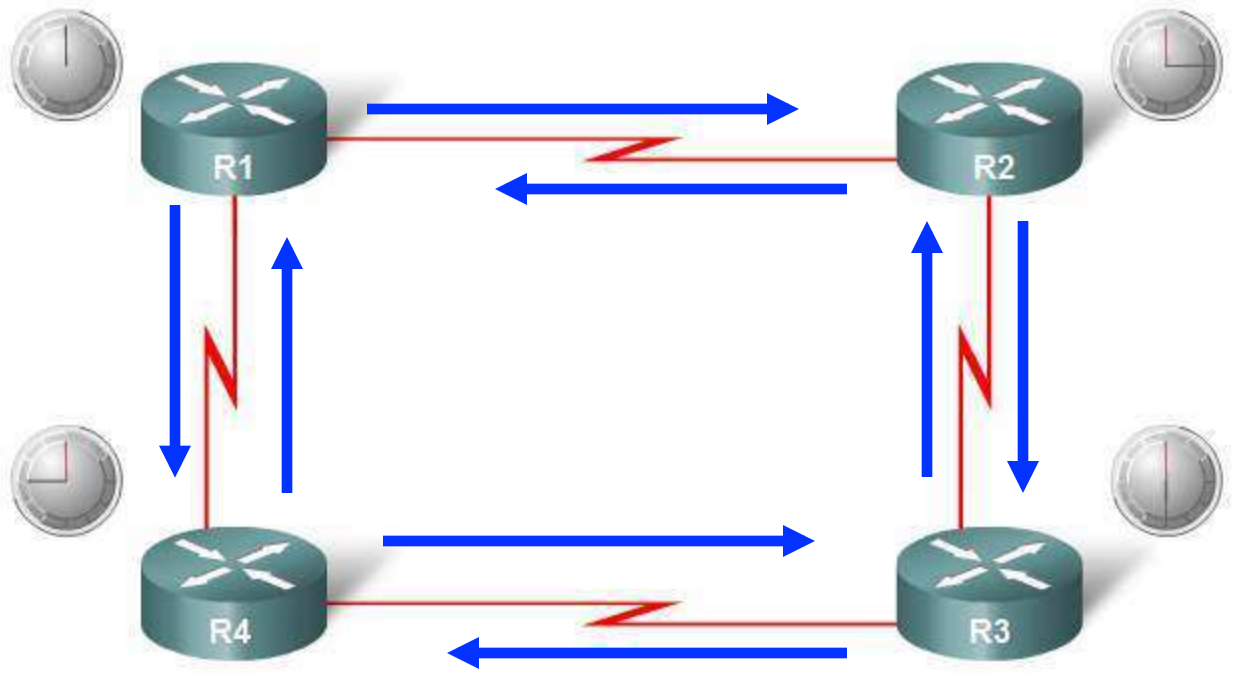
Inefficace :

Gaspillage de bande passante et de CPU

Envoi régulier même si aucun changement a eu lieu

Opération des protocoles à vecteur de distances

Timer Expires



Les protocoles de routage utilisent

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

Mises à jour par **Multicast**

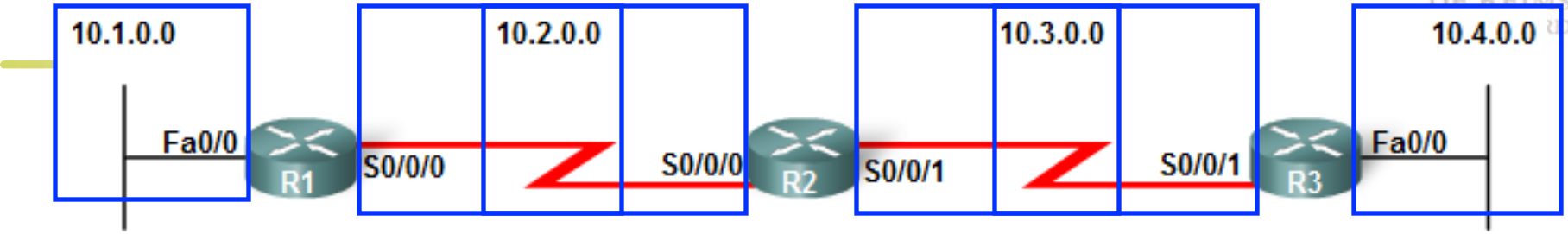
Les voisins doivent traiter les mises à jour

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

Découverte d'un Réseau

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

Cold Start



Network	Interface	Hop

Network	Interface	Hop

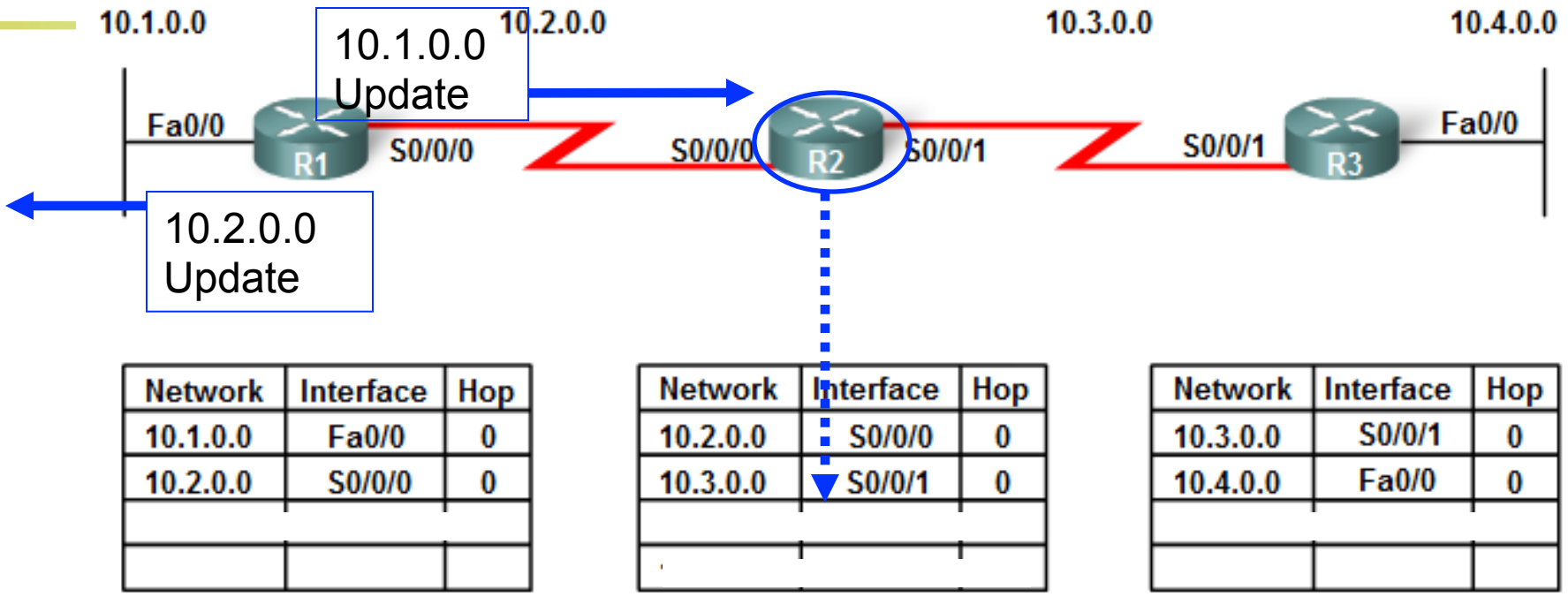
Network	Interface	Hop

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

Note : La totalité des tables de routage est envoyé

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

Échange Initial



Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0

Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0

Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0

R1: envoie tout sa table de routage

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

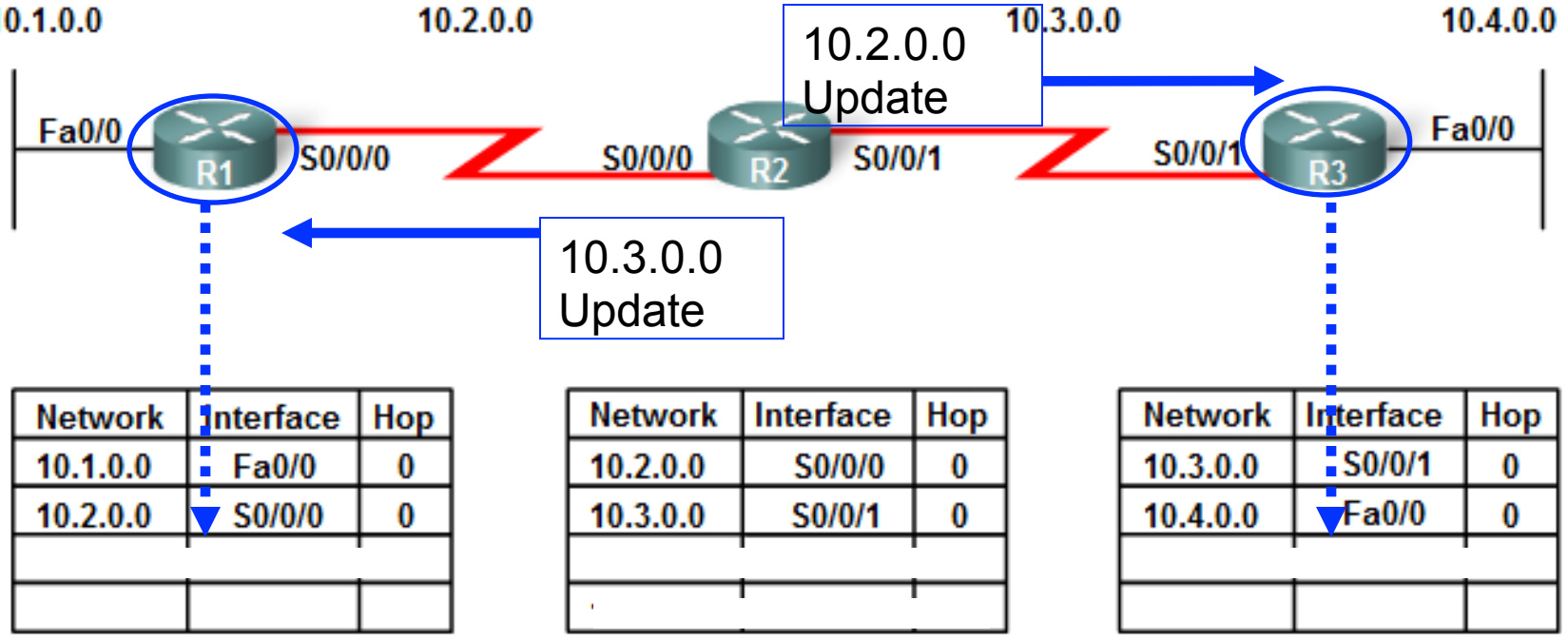
Envoie une mise à jour via FastEthernet 0/0 avec une métrique 1

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

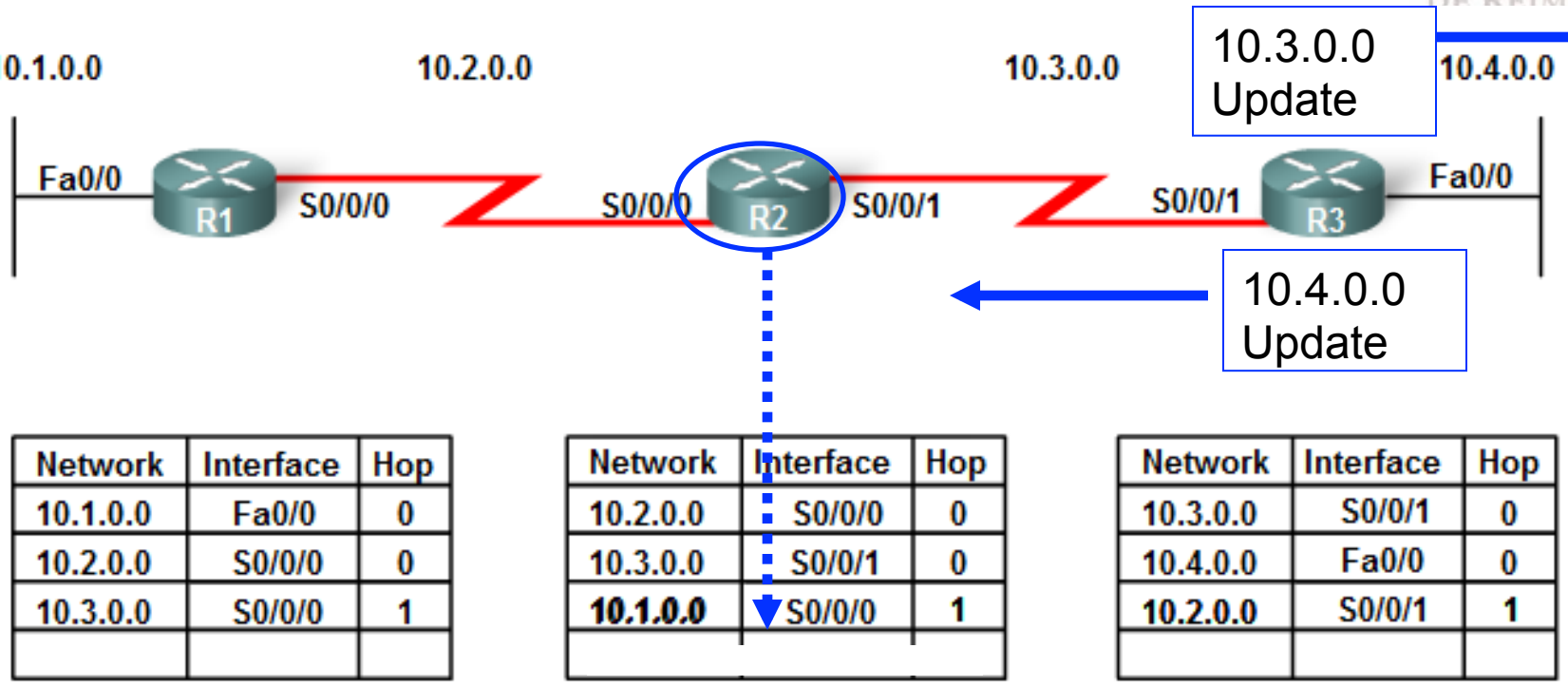
R1 **Reçoit** la mise à jour de R2 sur 10.3.0.0 sur son Serial 0/0/0 avec une métrique 1

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



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

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

R3: (Au même temps que R1 et R2)

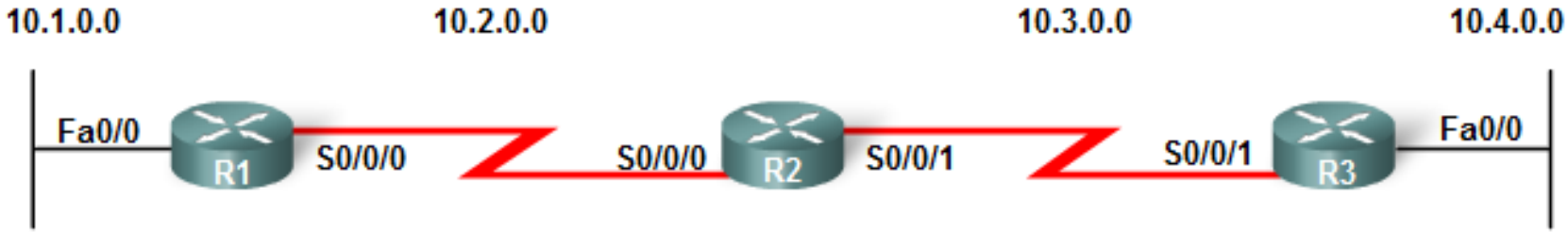
Envoie une mise à jour sur 10.4.0.0 via Serial 0/0/1 avec une métrique 1

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
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

Le réseau a-t-il convergé ?

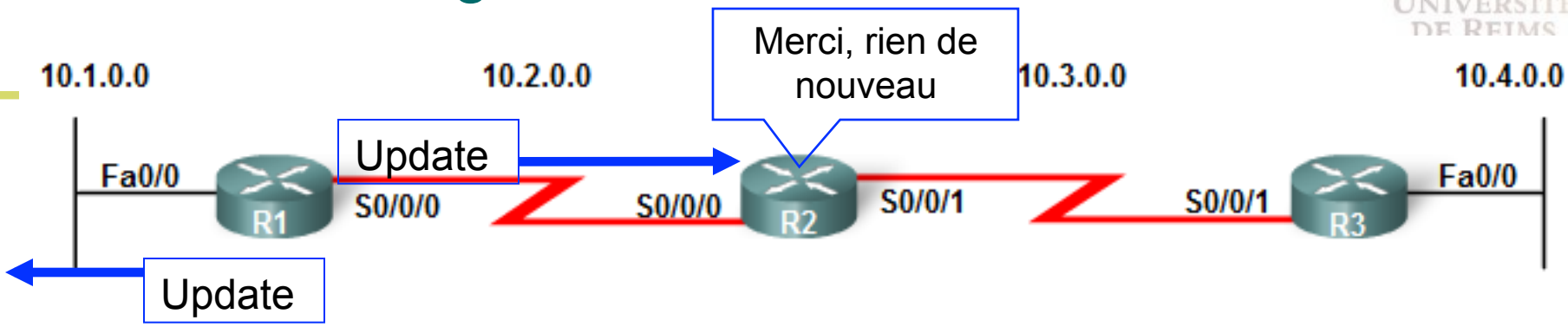
Non

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

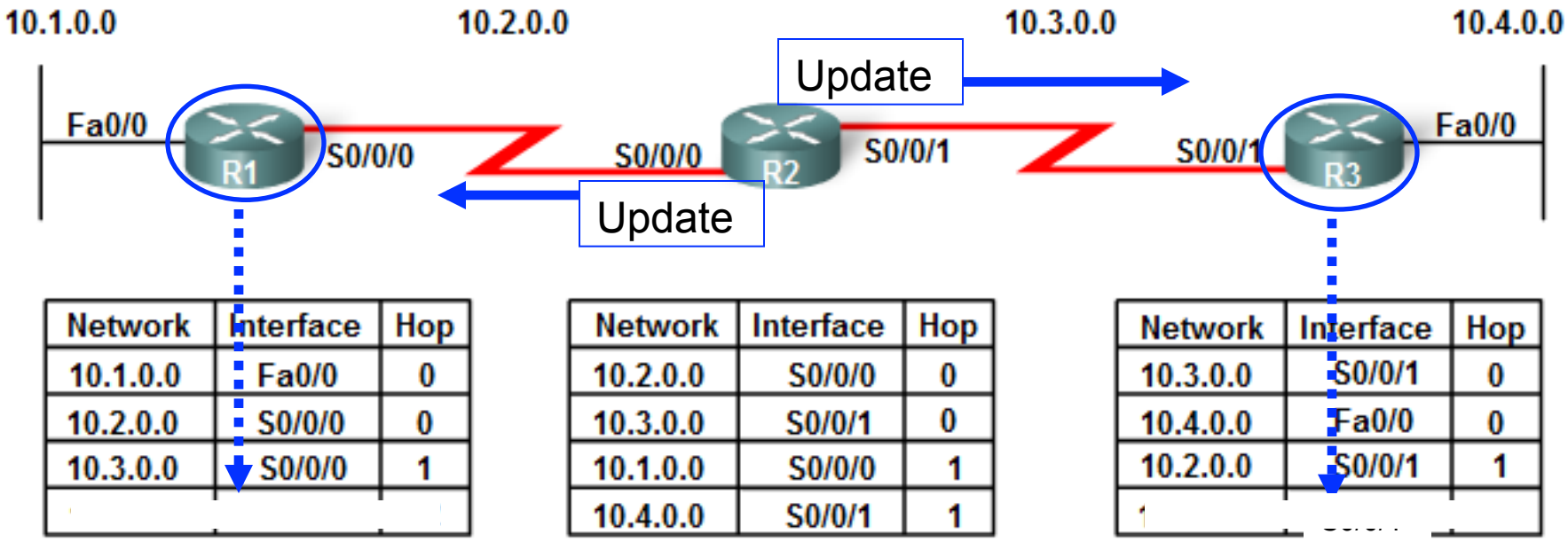
R1:

Envoie la totalité de sa table de routage

R2 apprend quelque chose nouvelle ?

Non

Prochain Échange



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

R2:

Envoie la totalité de sa table de routage

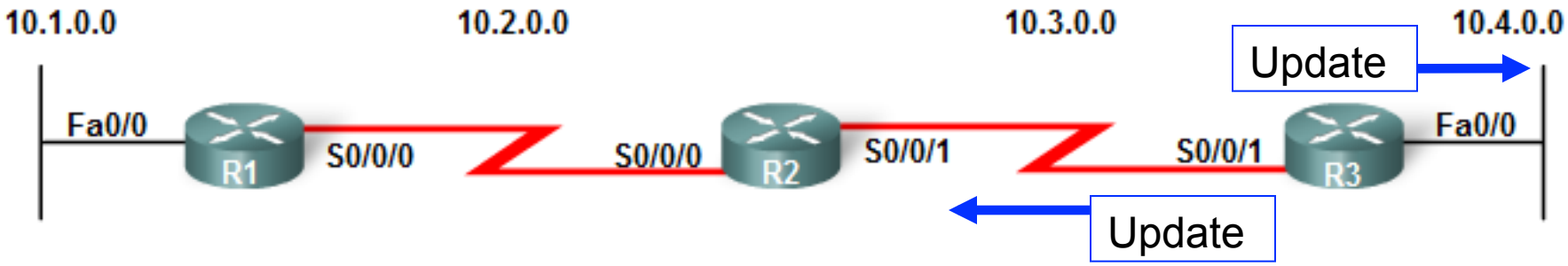
R1 apprends quelque chose nouvelle ?

Oui, 10.4.0.0

R3 apprends 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

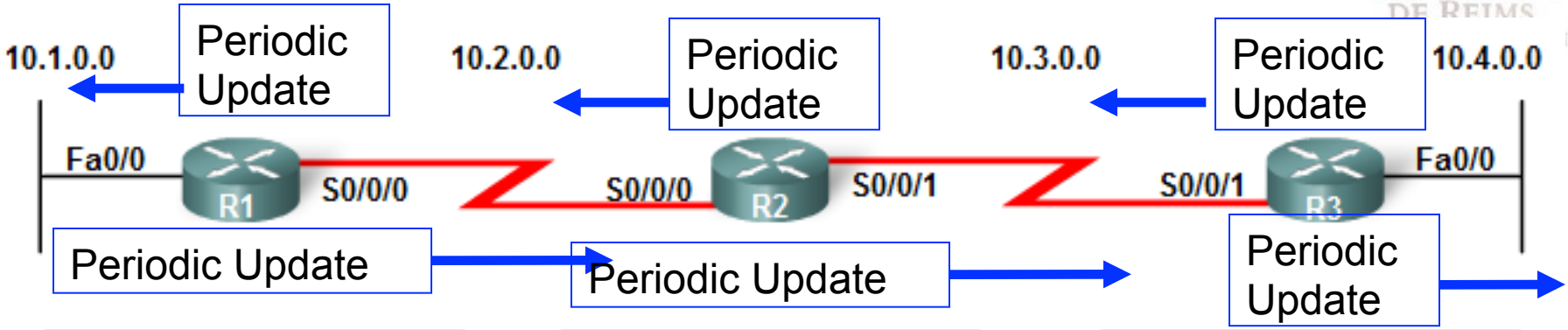
R3:

Envoie la totalité de sa table de routage

R2 apprend quelque chose nouvelle ?

Non

Mises à jour périodiques



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

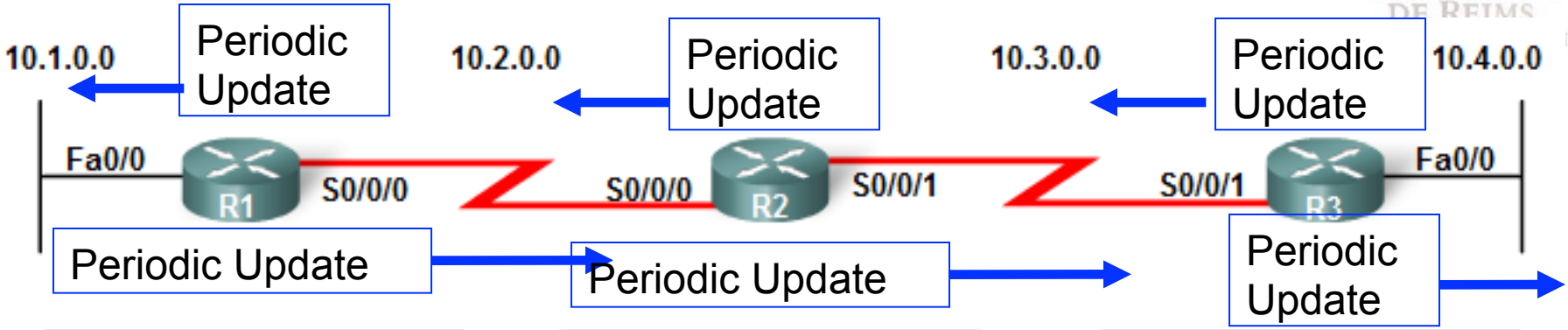
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

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

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

La plupart des protocoles à vecteur de distances (dont RIP) choisit d'envoyer régulièrement sa table de routage, même quand aucune modification a été observée

Mises à jour périodiques



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

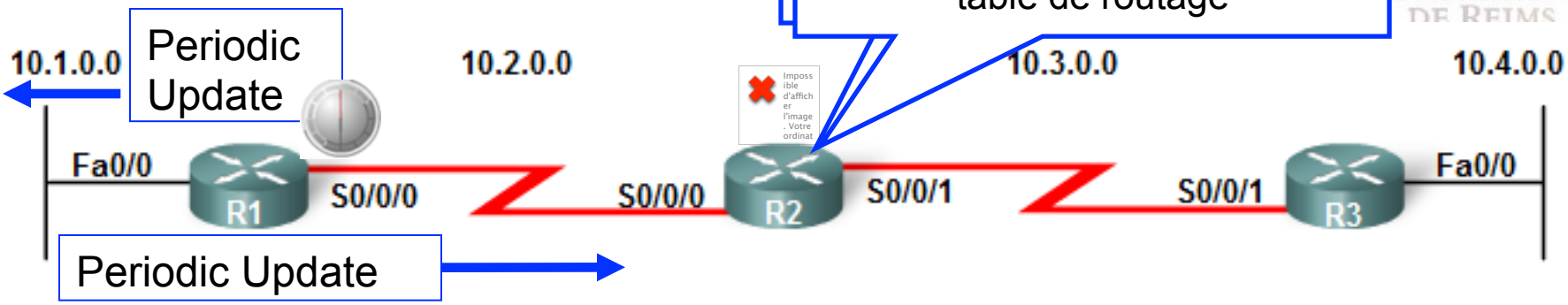
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

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

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



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 but de180 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) - plus tard

Temporisateurs RIP

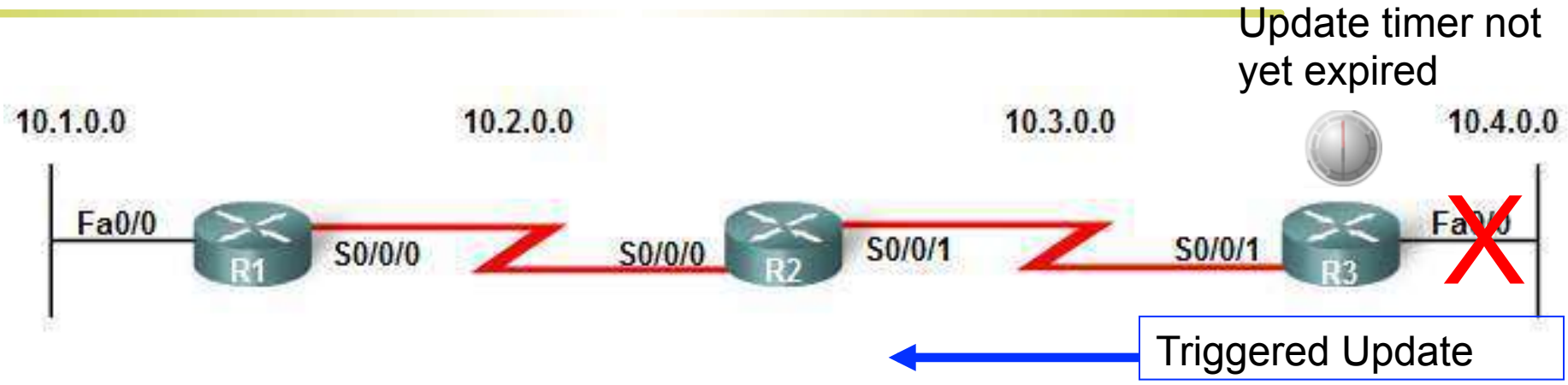

```
R1# show ip route

     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`**

Mises à jour déclenchées (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

Inconvénients des Protocoles à Vecteur de Distances

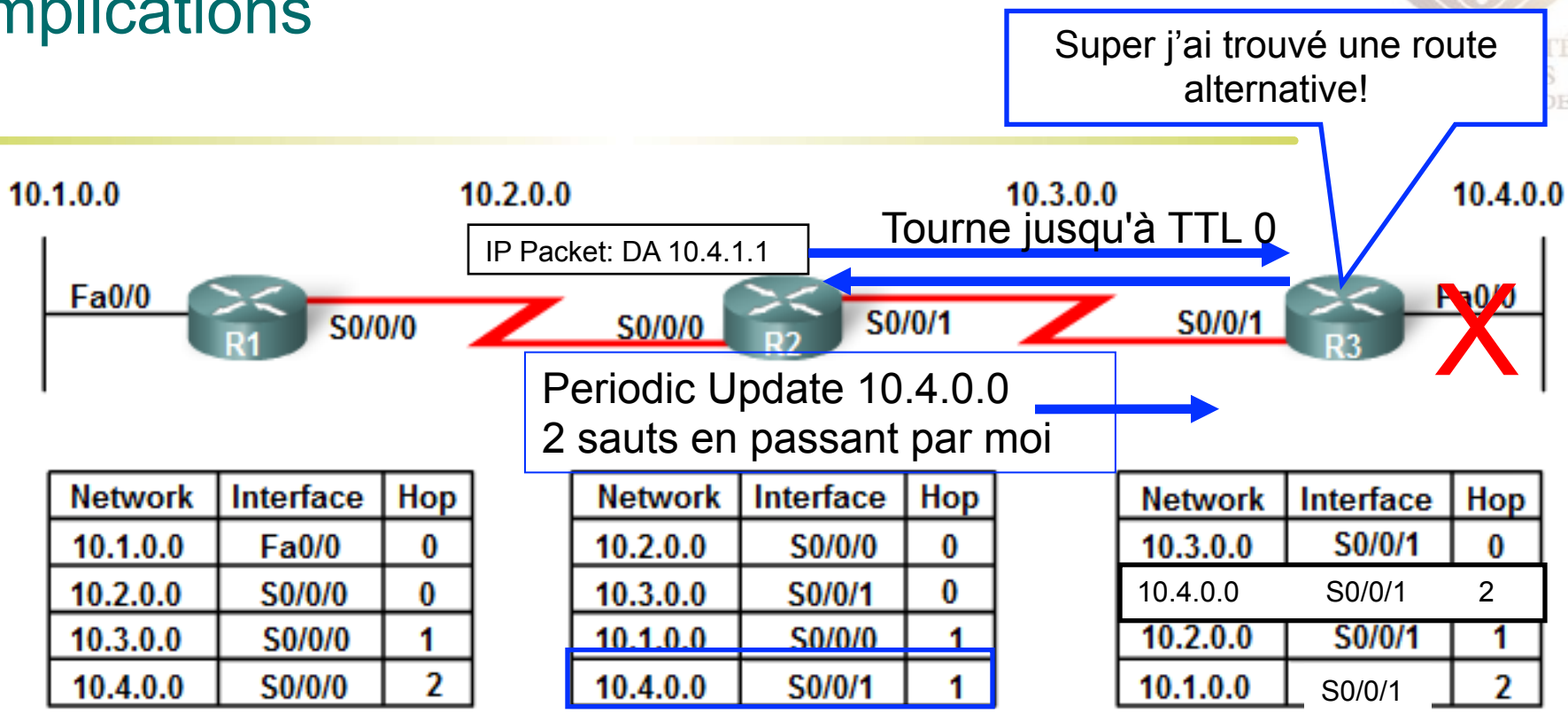
Boucle de routage

- Une **boucle de routage** est une condition dans laquelle un paquet est continuellement relayé par une série de routeurs 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 distances
- La boucle peut être causée par :
  - Des routes statiques mal configurées
  - Routage contradictoire à cause d'une convergence lente
  - Redistribution incorrecte de routes

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



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

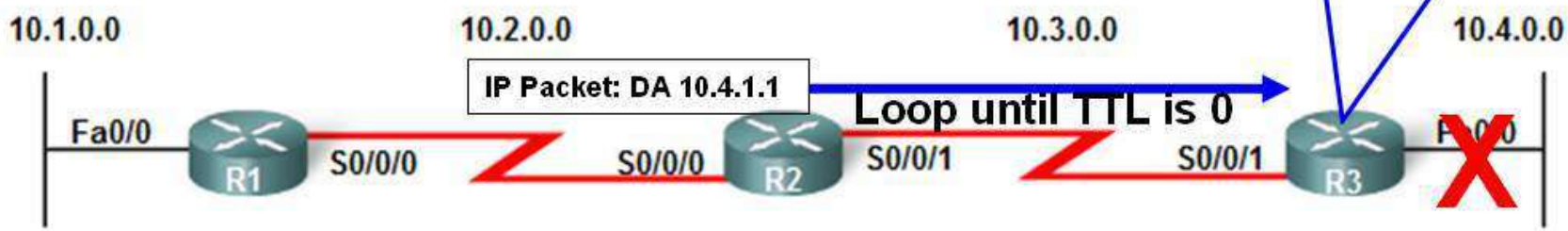
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

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

Que arrive-t-il 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

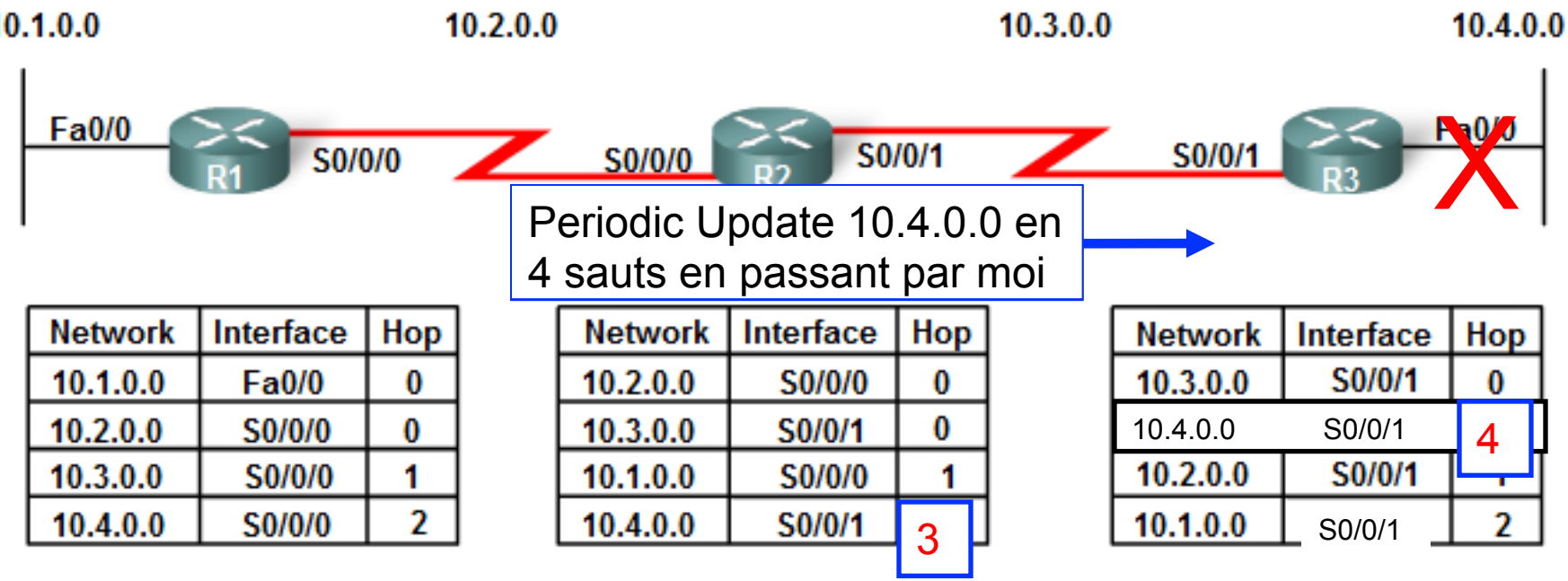
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

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 le Split horizon
- Faire de la route avec retour empoisonnée (poison reverse)
- Utiliser des Hold-down timers
- Des mises à jour déclenchées (déjà vu)

Compter à l'infini



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

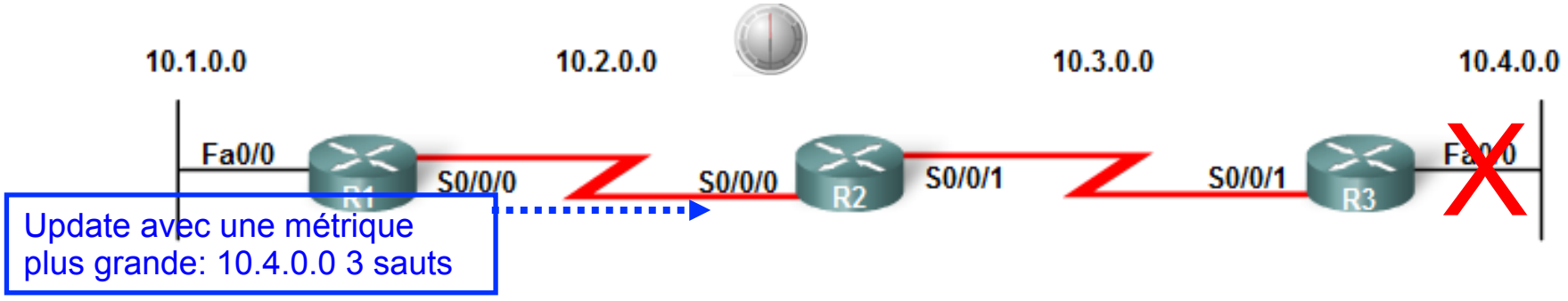
Chaque protocole a une valeur différente

RIP définit 16 comme "infini"

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

Horizon partagé – split-horizon

- Dans l'exemple précédent on a vu que l'un des problèmes était la "réalimentation" des routes par des nœuds distants
- Pendant qu'on compte "à l'infini", les paquets tournent en rond entre les machines
- Il faut trouver un moyen de bloquer cette réalimentation

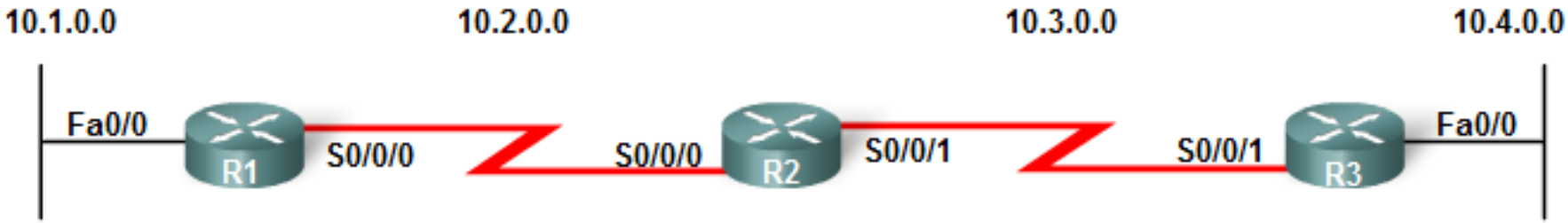


Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

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



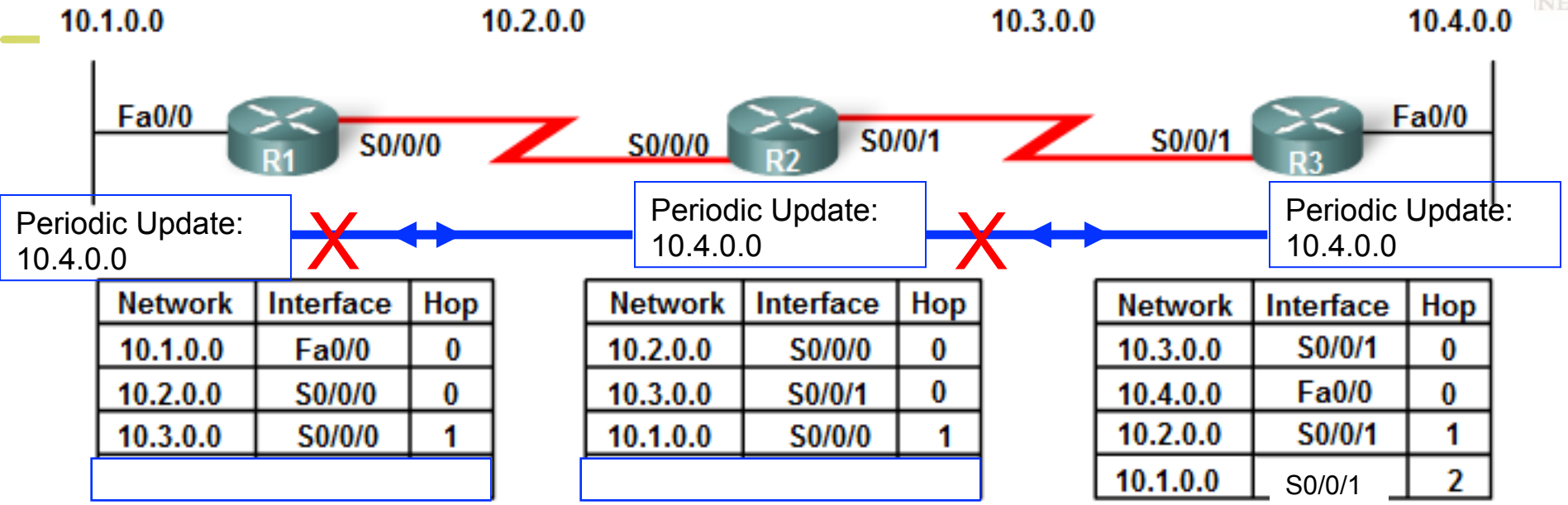
Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

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** et une technique qui empêche une route d'être annonce dans la direction d'où elle a été apprise
- Empêche que les routeurs soient "réalimentés" avec des informations anciennes

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

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

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

1. R3 annonce le réseau 10.4.0.0 à R2

2. R2 reçoit l'information et met à jour sa table de routage

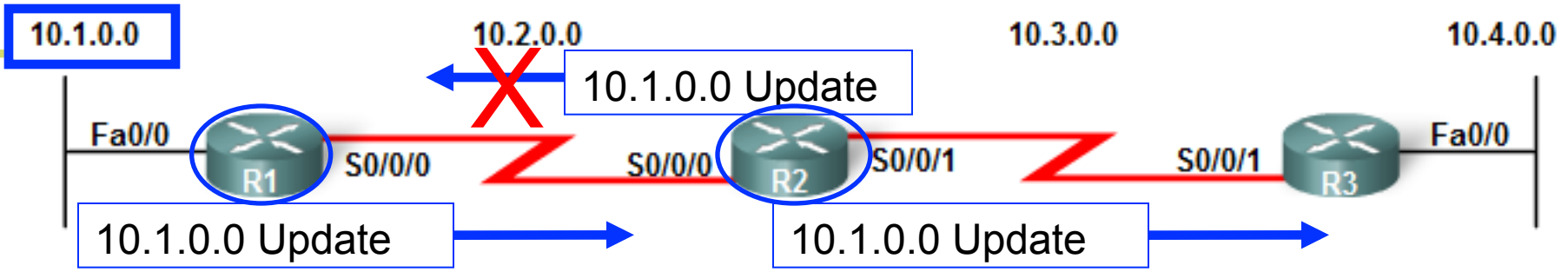
3. R2 annonce 10.4.0.0 à R1 via S0/0/0

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

Rappel sur le 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

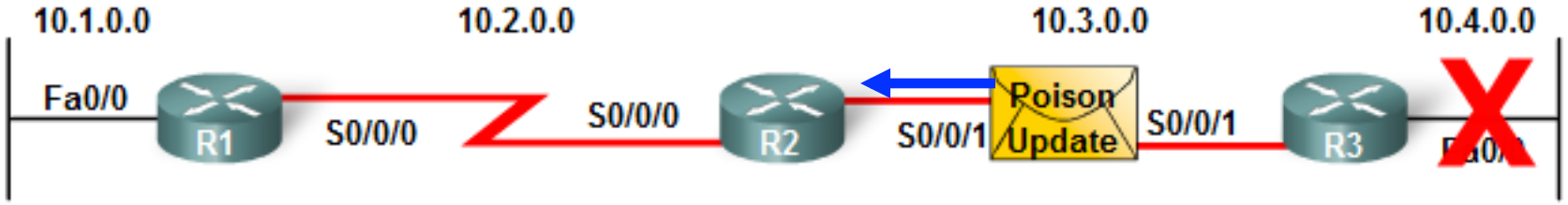
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

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

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'ou elle a été aprise

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.

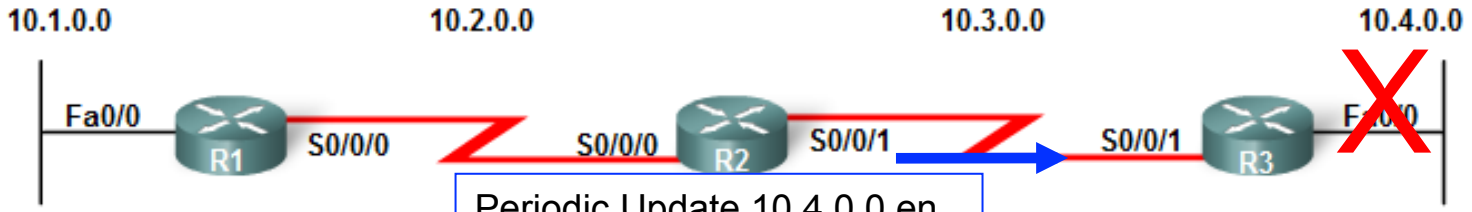
Inatteignable = métrique au maximum

Les routes empoisonnées accélèrent la convergence

Risque : si un update se perd, on peut trouver des boucles de routage dans d'autres parties du réseau

Split-horizon avec retour empoisonné

- Les deux techniques précédentes travaillent dans des directions différentes
  - Split-horizon : empêche le retour "→"
  - Route empoisonné : annonce des routes infinies "←"
- Si les deux techniques ne sont pas bien accordées, des boucles de routage peuvent toujours avoir lieu
  - Update empoisonné perdu, split-horizon en marche → comptage à l'infini
- Une manière d'empêcher ceci est de "mélanger" les deux approches



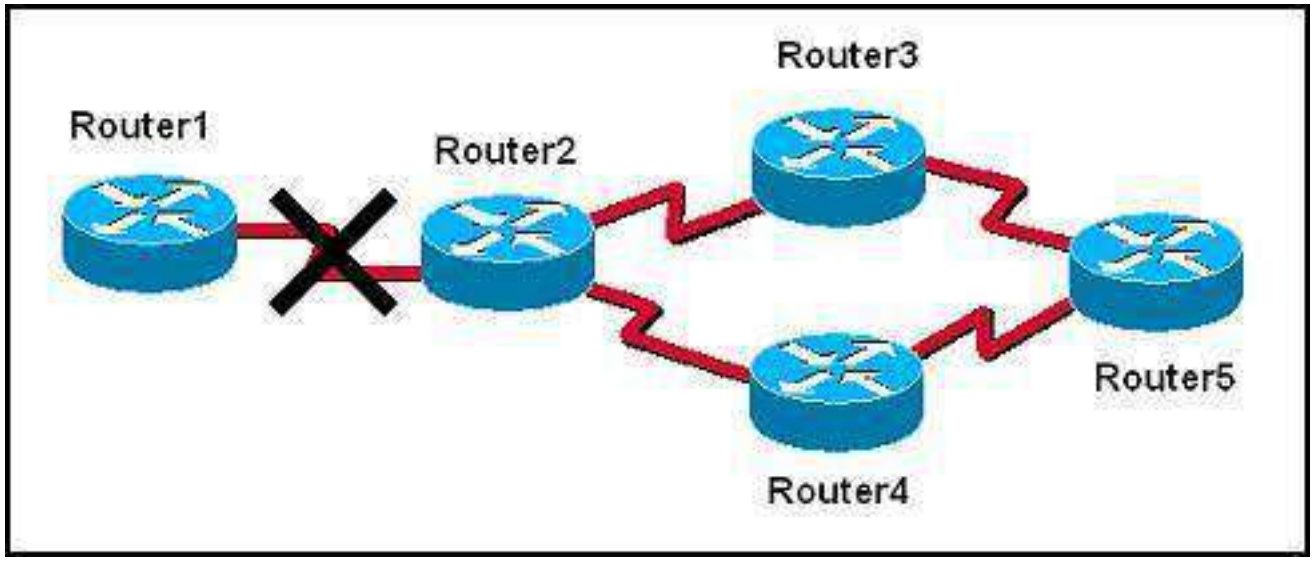
Network	Interface	Hop
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

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

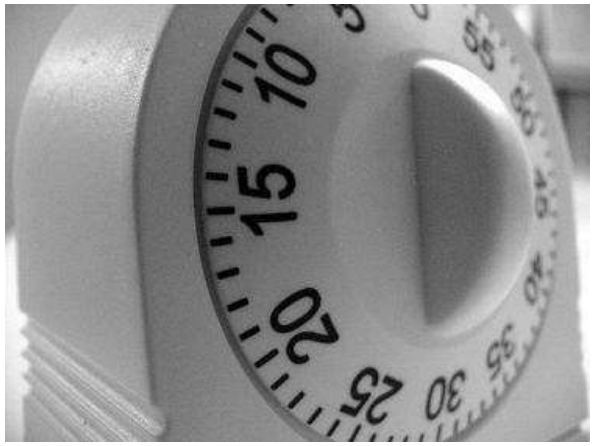
Network	Interface	Hop
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Est-ce fini ?

- Est-ce que le problème des boucles de routage est fini avec l'utilisation de l'horizon partagé avec retour empoisonné ?
- **NON**
- Ces techniques n'empêchent que les boucles entre deux nœuds
- Cet autre scénario n'est pas concerné :



Hold-Down Timers



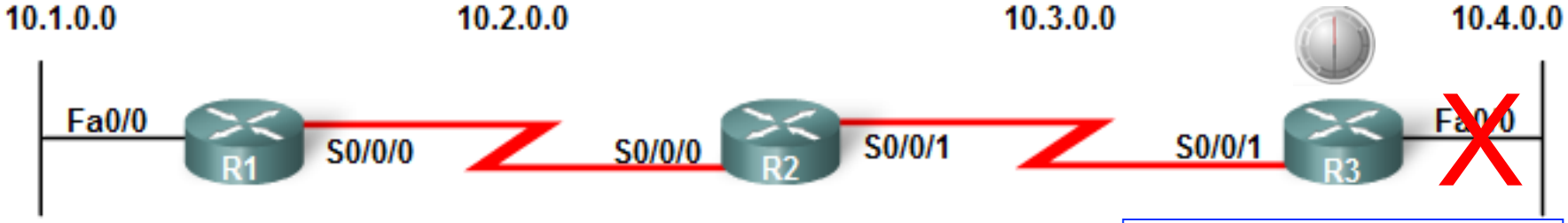
Une boucle de routage peut être crée si une mise à jour est reçue/envoyée en période de instabilité

Hold-down timers :

Empêchent les mises à jour dans les conditions de instabilité

Hold-Down Timers

Update timer not yet expired



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

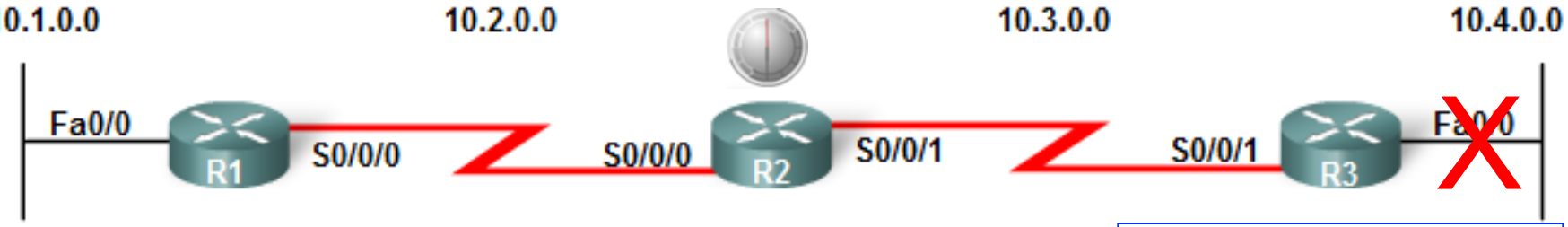
Network	Interface	Hop
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.1.0.0	S0/0/0	1
10.4.0.0	S0/0/1	1

Network	Interface	Hop
10.3.0.0	S0/0/1	0
~~10.4.0.0~~	~~Fa0/0~~	~~0~~
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

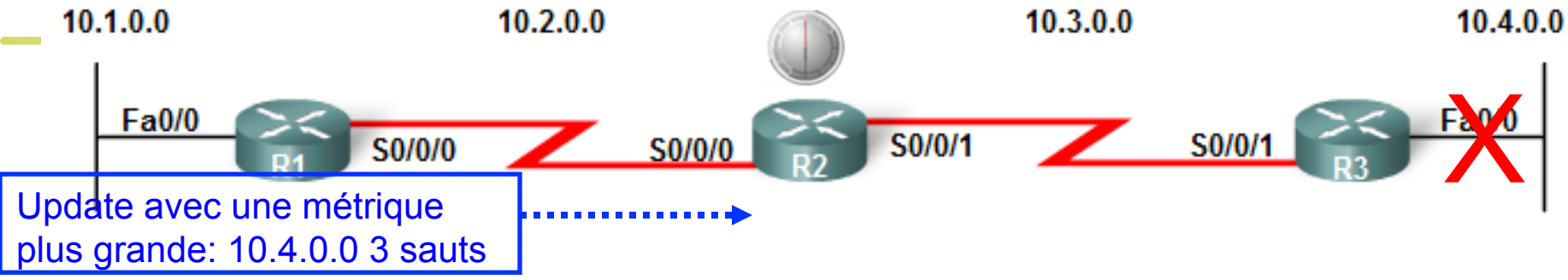
Network	Interface	Hop
10.3.0.0	S0/0/1	0
~~10.4.0.0~~	~~Fa0/0~~	~~0~~
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

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



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

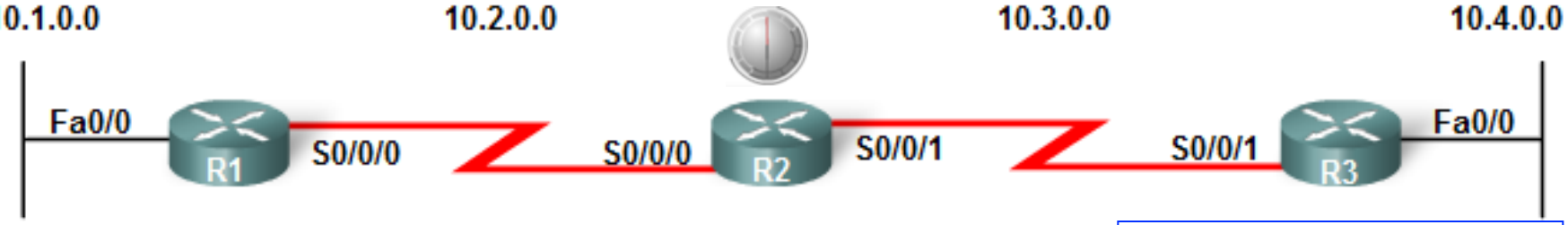
Network	Interface	Hop
10.3.0.0	S0/0/1	0
~~10.4.0.0~~	~~Fa0/0~~	~~0~~
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

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

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
10.4.0.0	Fa0/0	0
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

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