

## Chapitre 3

# Routage Dynamique

Dans un environnement complexe, la mise en œuvre du routage statique est souvent difficile à maintenir.

Les conditions dans le réseau changent. Des liens peuvent tomber et des équipements peuvent être en panne. Il faut prendre en compte ces événements lors que l'on construit les routes. Ainsi, on fera du routage dynamique. Les routeurs échangent des informations pour construire une connaissance commune du réseau de façon distribuée et découvrir les chemins possibles.

La mise en place d'un mécanisme de routage dynamique permet de faciliter les mises à jour.

Chaque routeur diffuse la liste des réseaux sur lesquels il est connecté.

Chaque routeur met à jour sa table de routage à partir des informations reçues depuis les autres.

Le routage dynamique apporte une certaine robustesse face aux pannes. En revanche, il apporte un problème puisque, lorsqu'on va recalculer la route de détour, il faudra faire attention à ne pas créer de **boucle**. En effet, une boucle sur un chemin fait que le trafic n'arrive pas à destination. De plus, on utilise inutilement la bande passante sur ce chemin.

Quand on va chercher à recalculer la nouvelle route, le réseau se trouve dans un état instable.

Pendant ce temps de **convergence**, il est possible que des boucles transitoires existent sur le réseau.

## 1. Protocoles de routage

Le protocole de routage consiste à définir comment sont échangées les informations de routage, et donc à :

- découvrir les autres routeurs du réseau
- construire les tables de routage
- maintenir les tables de routage à jour

### 1.1 Classification des algorithmes de routage

- **Globale**
  - Connaissance globale du réseau
  - Calcul centralisé ou distribué
  - Notion d'état de liens
- **Décentralisé**
  - Pas de connaissance globale
  - Par itération

## 1.2 AS (Autonomous System)

Un **Autonomous System** (abrégié **AS**), ou **système autonome**, est un ensemble de réseaux informatiques IP intégrés à Internet et dont la politique de routage interne (routes à choisir en priorité, filtrage des annonces) est cohérente.

Un AS est généralement sous le contrôle d'une entité (Autorité) ou organisation unique, typiquement un fournisseur d'accès à Internet.

Les ressources d'adressage et de routage de l'internet -constituées par les adresses IP et les numéros AS ont été réparties par l'IANA auprès de RIR ((Regional Address Registry). Les RIR répartissent ensuite ces ressources auprès de Local Internet Registries (LIR = Registres Internet Locaux) qui attribuent les adresses aux utilisateurs finaux.

ARIN pour les zones Amérique du Nord

AfriNIC pour l'Afrique

APNIC pour les zones Asie-Pacifique

LACNIC pour les zones Amériques du Sud - Caraïbes

RIPE NCC pour la zone Europe étendue.

Les numéros d'AS sont des entiers stockés sur 16 bits. Il ne peut donc y en avoir que 65535 au niveau mondial.

## 1.3 IGP (Interior Gateway Protocol)

IGP : Protocole de routage utilisé dans les réseaux sous même entité administrative

-qu'à l'intérieur d'une entité (entreprise, association, etc)

-décisions (suppression/ajout d'une ligne) peuvent être prises par un service unique

-Taille <100 routeurs, 1 autorité d'administration

Échange de routes, granularité = routeur

-but : trouver la route la plus efficace, en faisant confiance aux autres routeurs.

Exemples :

-RIP (Routing Information Protocol) v1, v2 : protocole à vecteur de distance (Distance Vector)

-OSPF (Open Shortest Path First) : protocole de routage à état de lien (Link-state)

-IGRP/EIGRP protocole propriétaire CISCO

## 1.4 EGP (Exterior Gateway Protocol)

EGP : Protocole de routage adapté à la redistribution de préfixes vers des réseaux extérieurs, ayant une entité administrative différente

-s'utilise entre entités distinctes (souvent concurrentes).

-Impossibilité de prendre une décision qui s'imposera à tous.

-On n'est pas prévenu de ce que vont faire les autres.

-Idéal de méfiance : le but n'est pas de trouver la meilleure route mais au contraire d'empêcher les routeurs de choisir une route dont on ne voudrait pas.

-Taille = Internet, coopération d'entités indépendantes

-Échange d'informations de routage, granularité = AS

-Pas d'information de routage mais d'accessibilité

Exemples :

-BGP (Border Gateway Protocol) : protocole à vecteur de chemin.

-C'est le protocole standard de l'Internet pour les interconnexions entre opérateurs.

## 2. Routage: principes de base

- Un algorithme de routage remplit une table de routage dans les routeurs

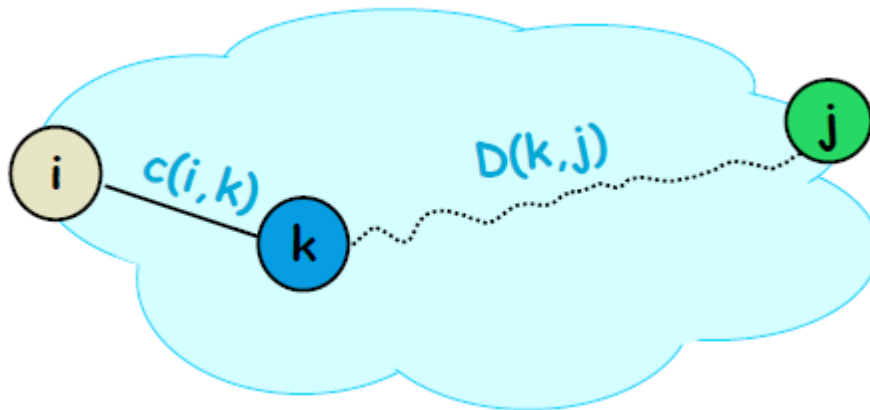
- Idée: échanger des messages pour connaître les voisins de ses voisins!

## 2.1 Critère de “consistance”

Un sous-ensemble d'un plus court chemin est aussi le plus court chemin entre les 2 deux nœuds intermédiaires

Corollaire:

- Si le plus court chemin d'un nœud  $i$  à un nœud  $j$ , avec une distance de  $D(i,j)$  passe par un nœud voisin  $k$  avec un lien de coût  $c(i,k)$ , alors:  $D(i,j) = c(i,k) + D(k,j)$



## Connaître les voisins de ses voisins...

- Un routeur connaît ses voisins directement connectés...
- ...ainsi que le “coût” pour y aller
- Coût = métrique pour comparer les routes entre elles
- La route la plus “courte” est celle dont le coût est minimal
- Voisin direct = 1 saut, donc coût de 1 si la métrique est le nombre de saut
- Voisin direct donc on connaît le débit du lien physique avec lequel on est connecté: débit faible=grand coût
- Par propagation des informations de voisinage et de coût, chaque routeur va construire une table de routage

## 2.2 Notion de convergence

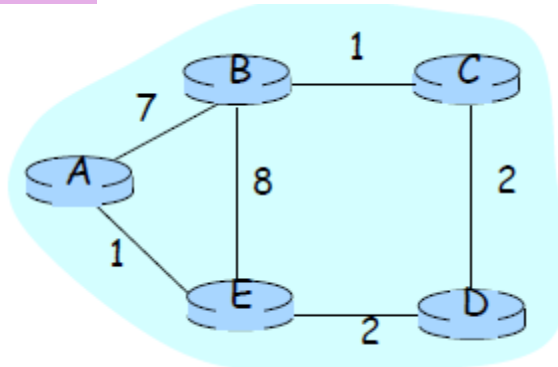
- On parle de convergence lorsque tous les routeurs ont la même information de routage
- En cas de non convergence, les ressources du réseau peuvent être inaccessibles
- Les paquets sont acheminés vers d'autres destinations.
- On parle de trou noir “Black holes” (les paquets disparaissent)
- Bouclage du processus de routage (routing loops)
- Le processus de convergence est déclenché après changement d'état d'un routeur ou d'un lien.

### 3. L'approche vecteur de distance

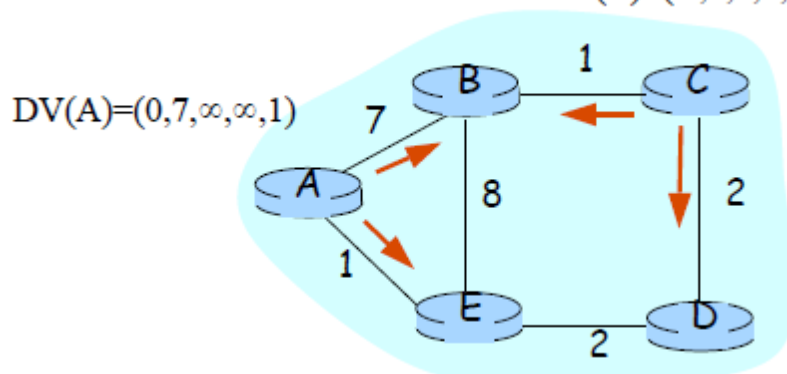
#### 3.1 Vecteur de distance (Distance-Vector, DV)

- chaque routeur ne connaît initialement que le coût de ses propres liaisons vers ses voisins direct. C'est le vecteur initial
- chaque routeur va échanger son vecteur initial avec tous ses voisins
- après un certain nombre d'itérations, chaque routeur va connaître le coût vers chaque destination,
- fonctionne bien sur des systèmes de petite taille.

$$DV(A)=(0,7,\infty,\infty,1)$$



$$DV(C)=(\infty,1,0,2,\infty)$$

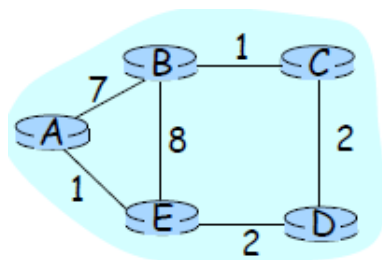


Pas obligatoirement de synchronisation dans les envois de messages

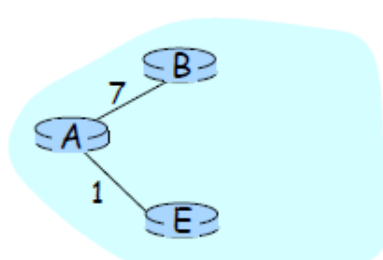
- **Condition de consistance:**  $D(i,j) = c(i,k) + D(k,j)$

- **L'algorithme DV (Bellman-Ford) évalue cette condition de manière récursive**

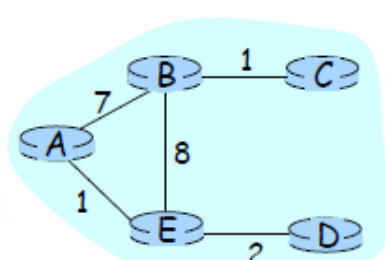
– À la m-ième itération, le critère de consistance est vérifié, en supposant que chaque noeud N "voit" les noeuds et les liens à msauts (ou moins) de lui (i.e. on a une vision à m-sauts)



Réseau d'étude



Vision de A à 1-saut  
(après la 1<sup>ère</sup> itération)



Vision de A à 2-sauts  
(après la 2<sup>nd</sup> itération)

### 3.2 Algorithme DV

#### - Valeurs initiales (itération 1):

- $D(i,i) = 0$  ;
- $D(i,k) = c(i,k)$  si  $k$  est un voisin (i.e.  $k$  est à 1 saut); et
- $D(i,j) = \text{INFINITY}$  pour tous non-voisin  $j$ .

#### - $D(i,*)$ est le vecteur de distance du noeud $i$ .

- L'algorithme maintient une table de "forwarding" pour toute destination  $j$ , initialisée comme suit:

- $\text{Prochain-saut}(i) = i$ ;
- $\text{Prochain-saut}(k) = k$  si  $k$  est un voisin, et
- $\text{Prochain-saut}(j) = \text{INCONNU}$  si  $j$  n'est pas un voisin.

- Après chaque itération chaque noeud  $i$  échange son vecteur de distance  $D(i,*)$  avec ses voisins directs.

- Pour tout voisin  $k$ , si  $c(i,k) + D(k,j) < D(i,j)$ , alors:

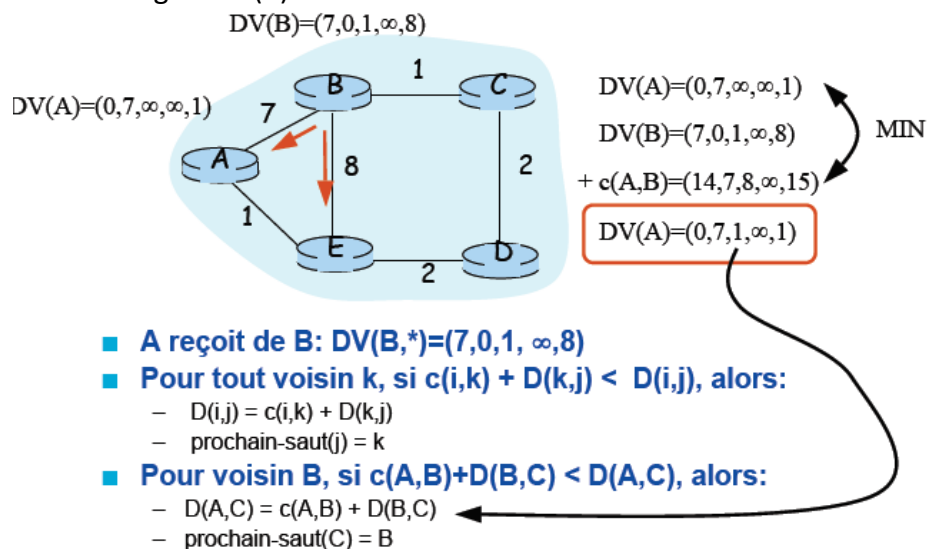
- $D(i,j) = c(i,k) + D(k,j)$
- $\text{prochain-saut}(j) = k$

- Après chaque itération, le critère de consistance est vérifié

– Après  $m$  itérations, chaque noeud connaît le plus court chemin vers tous les autres noeuds situés à  $m$  sauts ou moins.

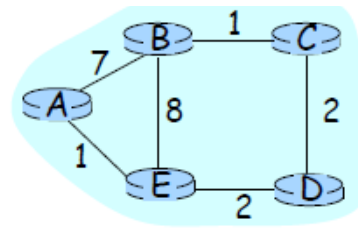
– i.e. chaque noeud a une vision à  $m$  sauts du réseau.

– L'algorithme converge en  $O(d)$  itérations:  $d$  est le diamètre maximum du réseau.

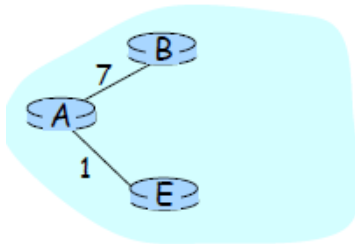


#### ■ Vecteur de distance de A: $D(A,*)$ :

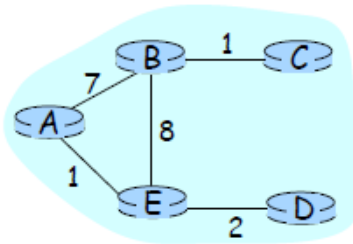
- Après Itération 1 est: [0, 7, INFINITY, INFINITY, 1]
- Après Itération 2 est: [0, 7, 8, 3, 1]
- Après Itération 3 est: [0, 7, 5, 3, 1]
- Après Itération 4 est: [0, 6, 5, 3, 1]



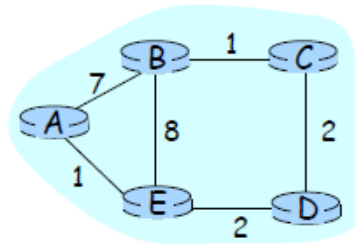
Réseau d'étude



Vision de A à 1-saut  
(après la 1<sup>ère</sup> itération)



Vision de A à 2-sauts  
(après la 2<sup>nd</sup> itération)



Vision de A à 3-sauts  
(après la 3<sup>e</sup> itération)

### 3.3 Problèmes avec le Distance Vector

#### 3.3.1 Boucle de routage

Une boucle de routage est une condition dans laquelle un paquet est transmis en continu entre une série de routeurs sans jamais atteindre le réseau de destination souhaité. Une boucle de routage peut se produire lorsque deux routeurs ou plus possèdent des informations de routage qui indiquent, à tort, qu'il existe un chemin valide vers une destination inaccessible.

La boucle peut être le résultat des problèmes suivants :

- Routes statiques configurées incorrectement
- Tables de routage incohérentes qui ne sont pas mises à jour en raison d'une convergence lente dans un réseau changeant
- Routes configurées ou installées incorrectement

**Remarque :** le protocole IP utilise son propre mécanisme pour empêcher la transmission sans fin d'un paquet sur le réseau. Le protocole IP possède un champ de durée de vie (TTL) dont la valeur est diminuée de 1 à chaque routeur. Si la durée de vie est égale à zéro, le routeur abandonne le paquet.

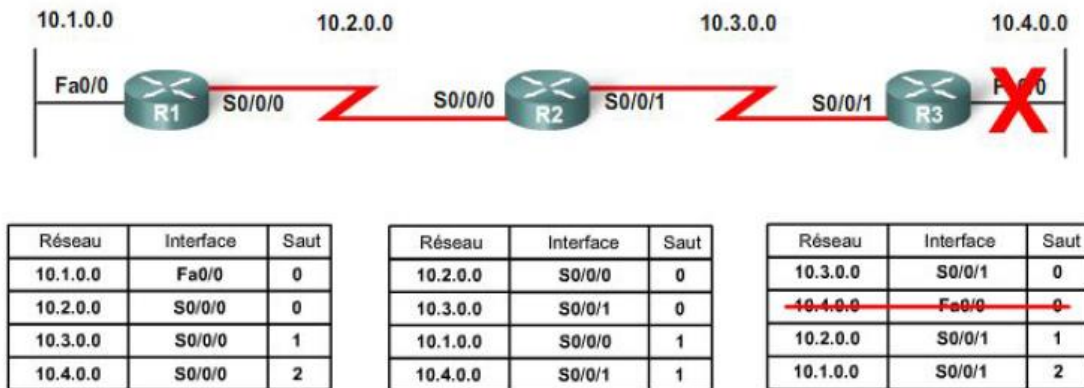
Une boucle de routage peut créer les conditions suivantes :

- La bande passante de la liaison est utilisée pour faire tourner le trafic en boucle entre les routeurs dans une boucle.
- Le processeur d'un routeur est fortement sollicité en raison des paquets tournant en boucle.
- Le processeur d'un routeur est surchargé en raison du réacheminement inutile de paquets, ce qui impacte négativement la convergence du réseau.

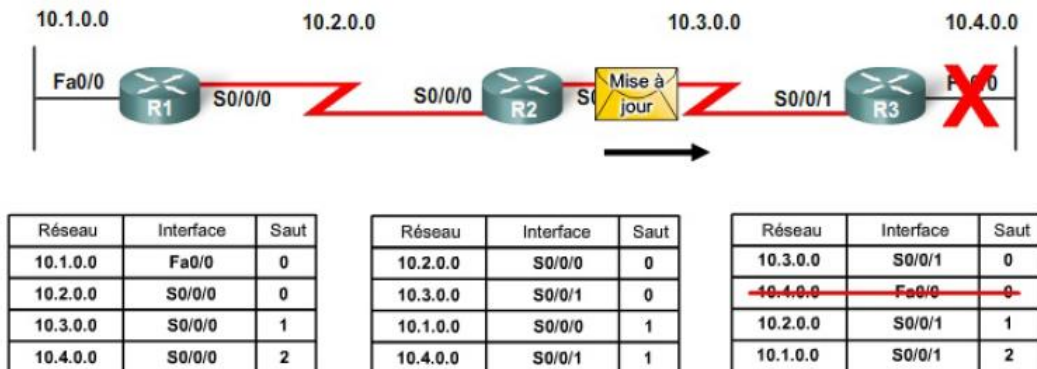
- Les mises à jour de routage peuvent se perdre ou ne pas être traitées en temps voulu. Ces conditions introduisent des boucles de routage supplémentaires qui aggravent davantage la situation.
- Les paquets peuvent se perdre dans des « trous noirs ».

### Exemples :

Le réseau 10.4.0.0 tombe en panne.

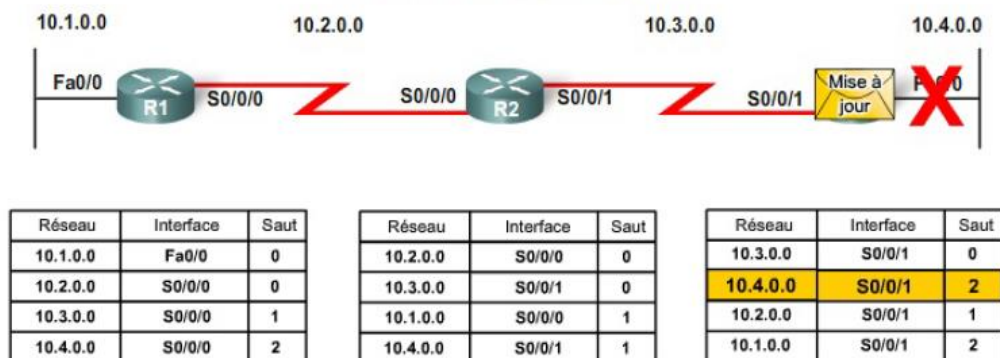


Mise à jour de routage de R2 vers R3



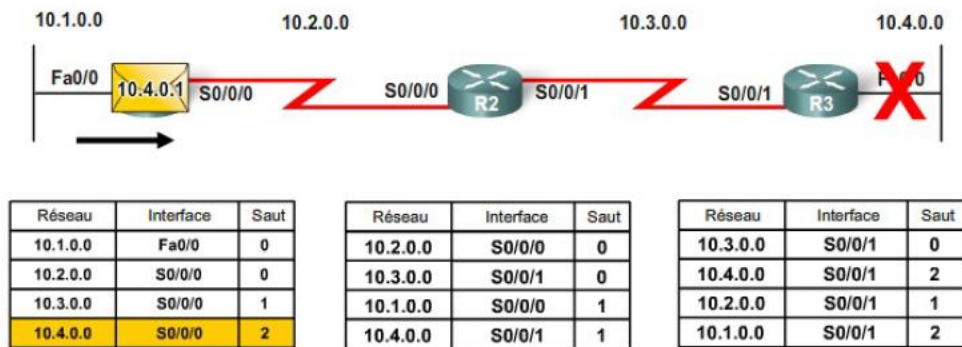
R3 installe une route « incorrecte » sur 10.4.0.0.

Le réseau possède maintenant une boucle.

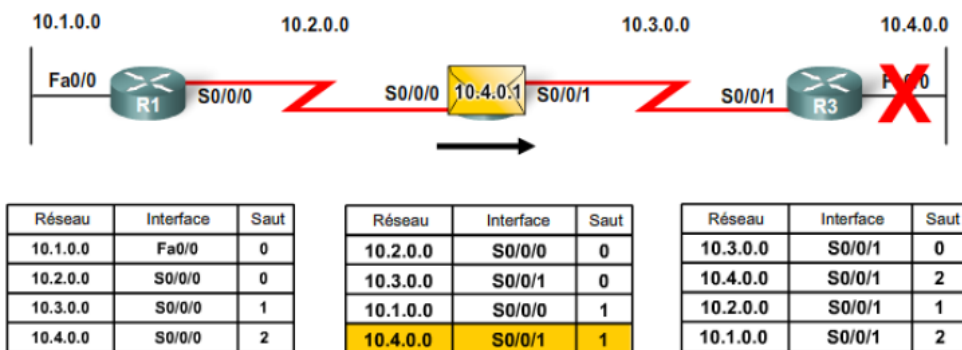




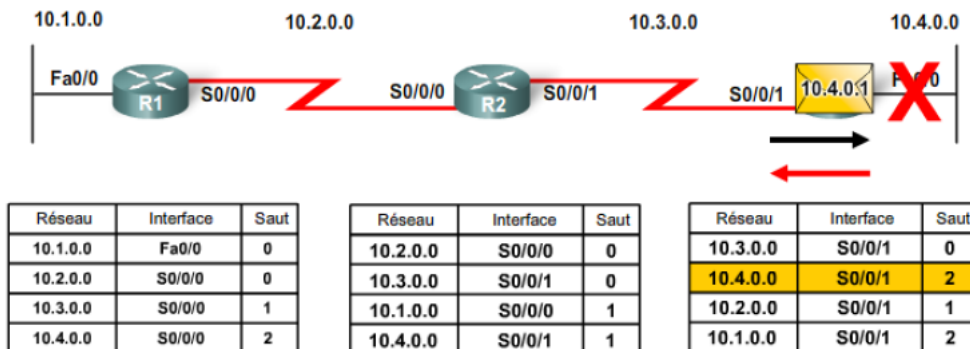
Le réseau possède maintenant une boucle.



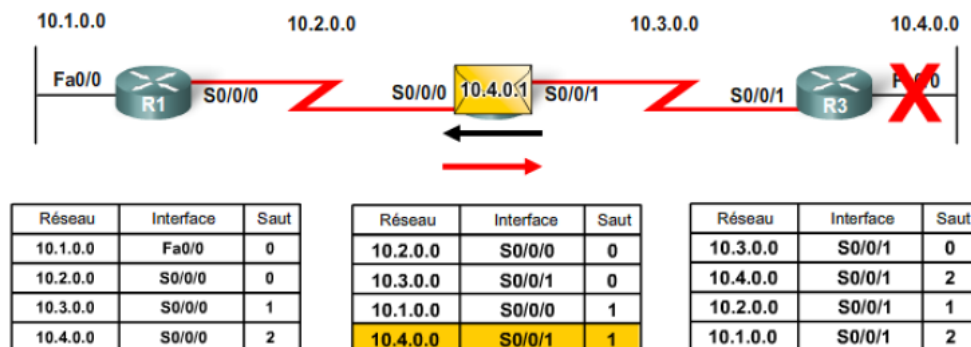
Le réseau possède maintenant une boucle.



Le réseau possède maintenant une boucle.



Le réseau possède maintenant une boucle.



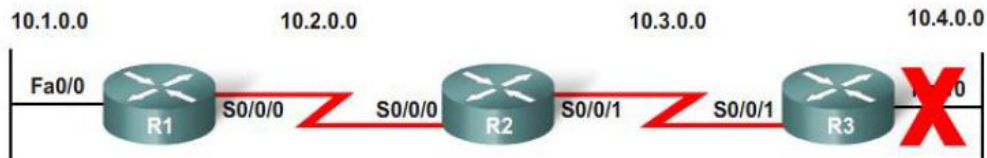


Le paquet sera détruit après un certain nombre d'échanges en boucle

### 3.3.2 Le comptage à l'infini

Le comptage à l'infini est une situation qui se produit lorsque des mises à jour de routage inexactes augmentent la valeur de la mesure jusqu'à l'infini pour un réseau qui n'est plus accessible.

Le réseau 10.4.0.0 tombe en panne.

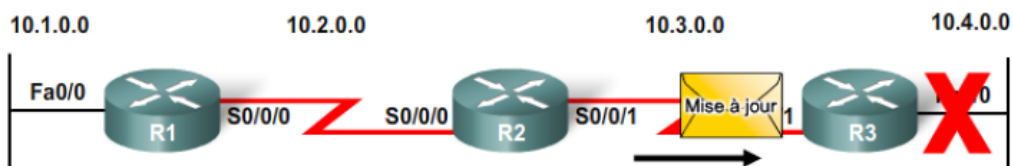


Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Réseau	Interface	Saut
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

Réseau	Interface	Saut
10.3.0.0	S0/0/1	0
<del>10.4.0.0</del>	<del>Fa0/1</del>	<del>0</del>
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

Pour que R3 puisse envoyer une mise à jour, R2 envoie une mise à jour.



Réseau	Interface	Saut
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

Réseau	Interface	Saut
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

Réseau	Interface	Saut
10.3.0.0	S0/0/1	0
<del>10.4.0.0</del>	<del>Fa0/1</del>	<del>0</del>
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

R3 installe une route « incorrecte » dans le réseau 10.4.0.0 avec un nombre de sauts égal à 2.

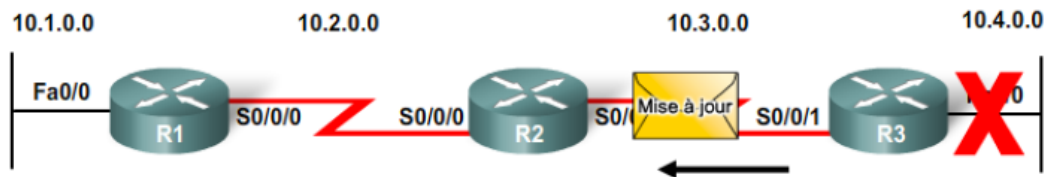


Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Réseau	Interface	Saut
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

Réseau	Interface	Saut
10.3.0.0	S0/0/1	0
10.4.0.0	Fa0/1	2
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

R3 envoie une mise à jour dans R2 avec un nombre de sauts égal à 3 dans le réseau 10.4.0.0.

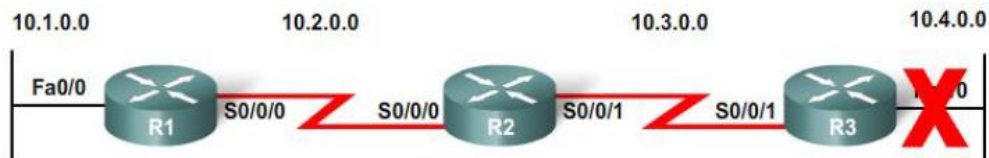


Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Réseau	Interface	Saut
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

Réseau	Interface	Saut
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

R2 fait passer le nombre de sauts du réseau 10.4.0.0 à 3.

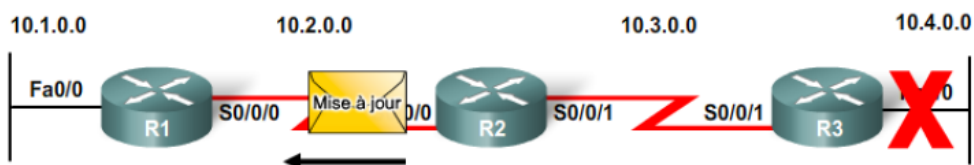


Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Réseau	Interface	Saut
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

Réseau	Interface	Saut
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

R2 envoie une mise à jour dans R1 avec un nombre de sauts égal à 4 dans le réseau 10.4.0.0.

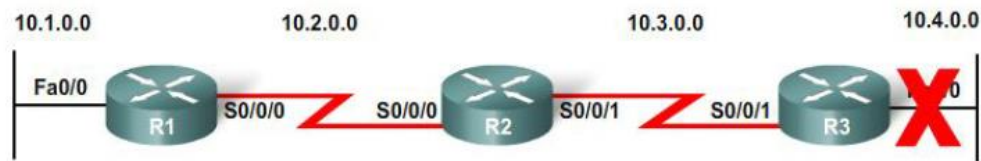


Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

Réseau	Interface	Saut
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

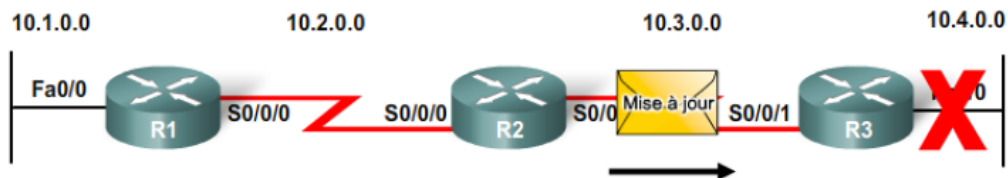
Réseau	Interface	Saut
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/1	2
10.2.0.0	S0/0/1	1
10.1.0.0	S0/0/1	2

R1 fait passer le nombre de sauts du réseau 10.4.0.0 à 4.



Réseau	Interface	Saut	Réseau	Interface	Saut	Réseau	Interface	Saut
10.1.0.0	Fa0/0	0	10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0
10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0	10.4.0.0	S0/0/1	2
10.3.0.0	S0/0/0	1	10.1.0.0	S0/0/0	1	10.2.0.0	S0/0/1	1
10.4.0.0	S0/0/0	4	10.4.0.0	S0/0/1	3	10.1.0.0	S0/0/1	2

R2 envoie la mise à jour périodique suivante avec un nombre de sauts égal à 4 dans le réseau 10.4.0.0.



Réseau	Interface	Saut	Réseau	Interface	Saut	Réseau	Interface	Saut
10.1.0.0	Fa0/0	0	10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0
10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0	10.4.0.0	S0/0/1	2
10.3.0.0	S0/0/0	1	10.1.0.0	S0/0/0	1	10.2.0.0	S0/0/1	1
10.4.0.0	S0/0/0	4	10.4.0.0	S0/0/1	3	10.1.0.0	S0/0/1	2

### 3.3.3 Solutions

Pour lutter contre les boucles de routage et comptage à l'infini, il y a des solutions par exemple :

- Définition d'une valeur de métrique maximale après laquelle le paquet sera détruit
- Minuteur de hors de service sur chaque routeur
- Découpage d'horizon
- Empoisonnement de routage

#### 3.3.3.1 Valeur maximale

Pour arrêter l'incrément d'une mesure, l'«infini» est défini par l'attribution d'une valeur maximale à la mesure. Par exemple, le protocole RIP considère que 16 sauts représentent l'infini, ce qui correspond à une mesure inaccessible. Une fois que les routeurs ont « compté jusqu'à l'infini », ils marquent la route comme étant inaccessible.

Les minuteurs (Timers) de mise hors service servent à empêcher les messages de mise à jour réguliers de rétablir de manière inappropriée une route qui se serait dégradée. Les minuteurs de mise hors service exigent des routeurs la suspension, pendant une durée spécifiée, de toute modification pouvant affecter les routes. Si une route est identifiée comme étant désactivée ou susceptible de l'être, toute autre information concernant cette

route ayant le même statut (ou un statut encore pire) est ignorée pendant une durée prédéterminée (la période de mise hors service).

Les minuteurs de mise hors service fonctionnent comme suit :

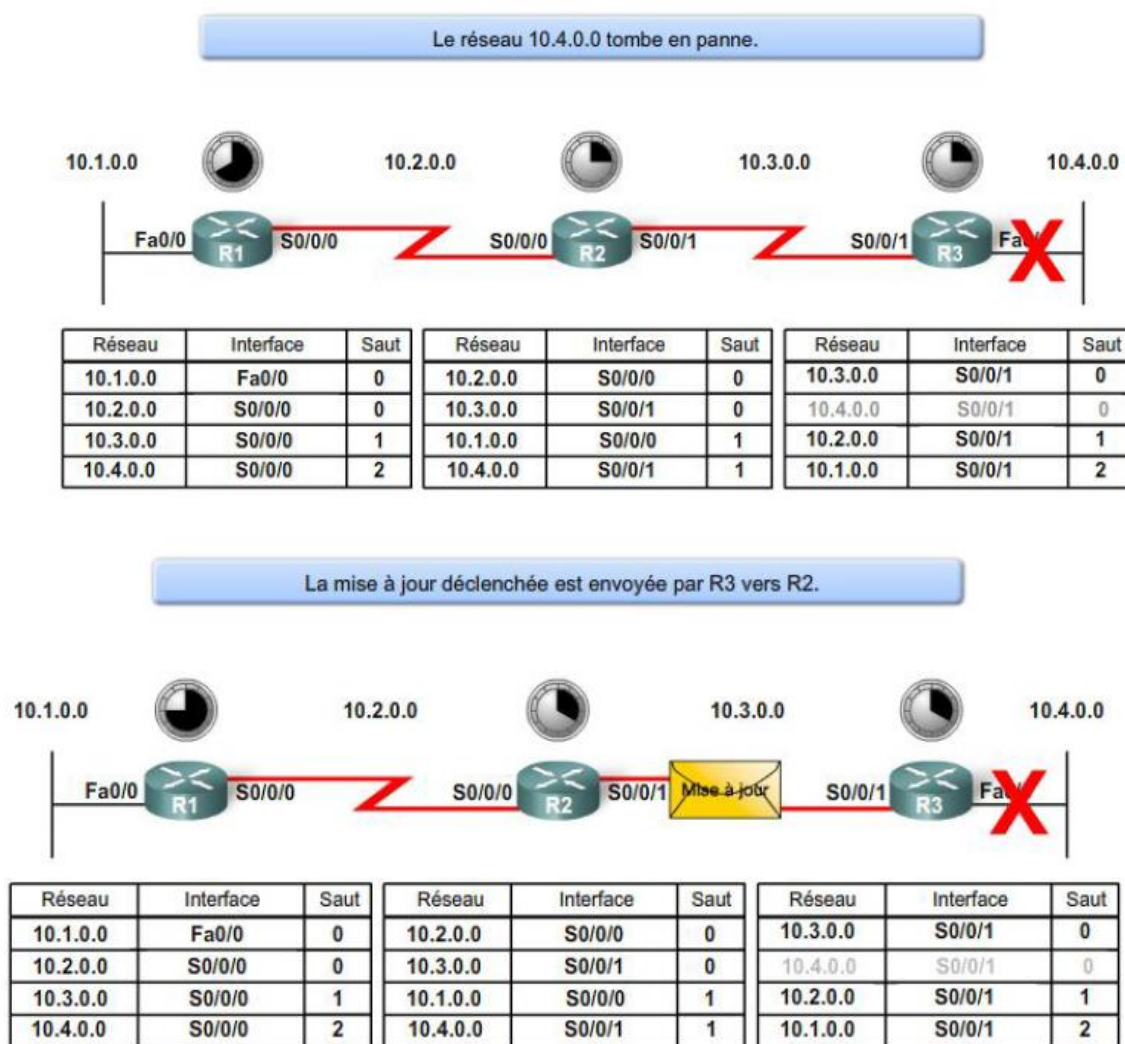
Un routeur reçoit une mise à jour d'un voisin lui indiquant qu'un réseau auparavant accessible est devenu inaccessible.

Le routeur marque la route comme étant éventuellement inactive et démarre le minuteur de mise hors service.

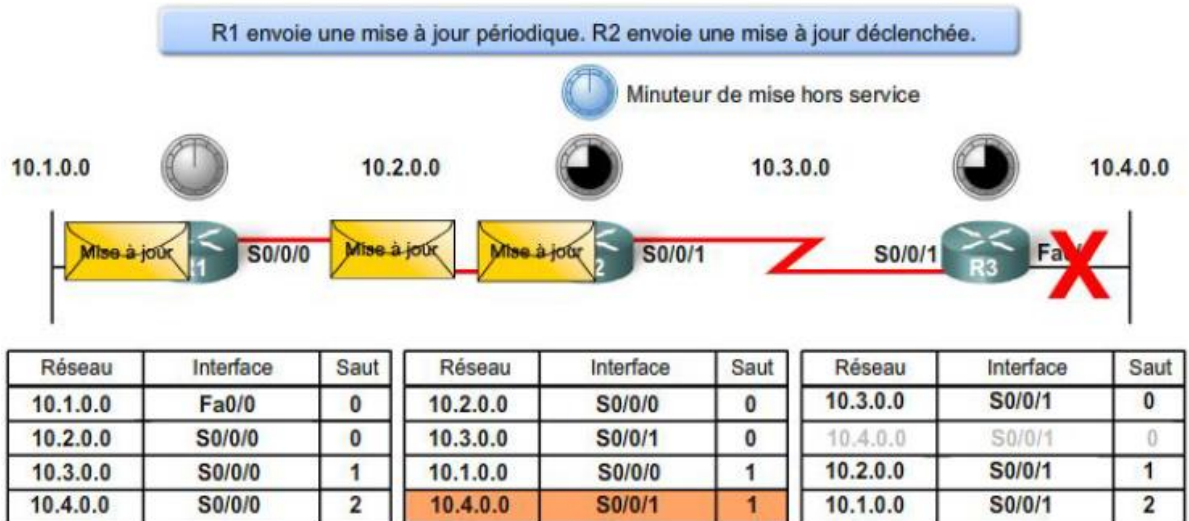
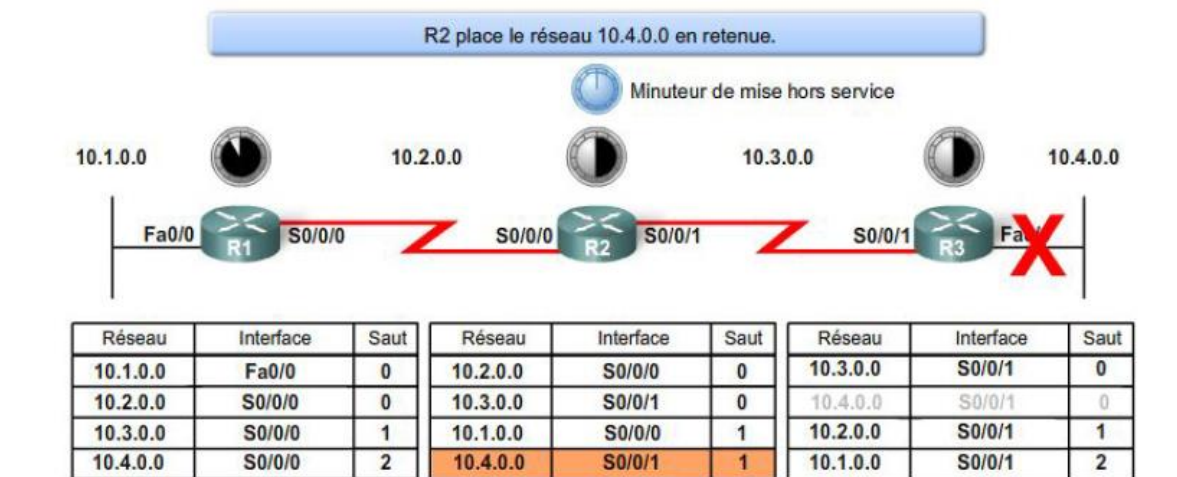
**Si une mise à jour avec une mesure inférieure pour ce réseau est reçue d'un routeur voisin au cours de la période de mise hors service, le réseau est rétabli et le minuteur de mise hors service est supprimé.**

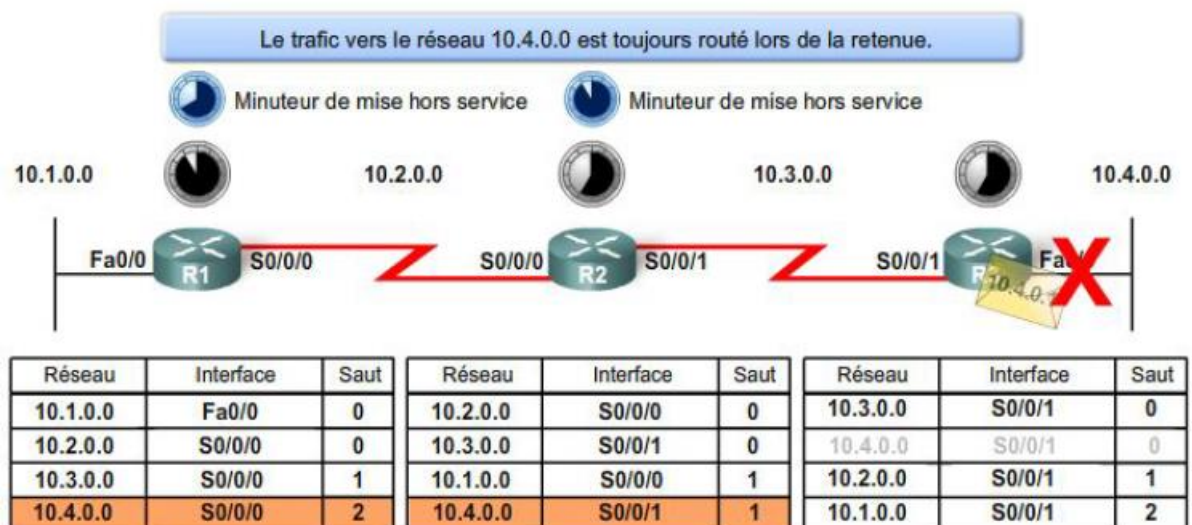
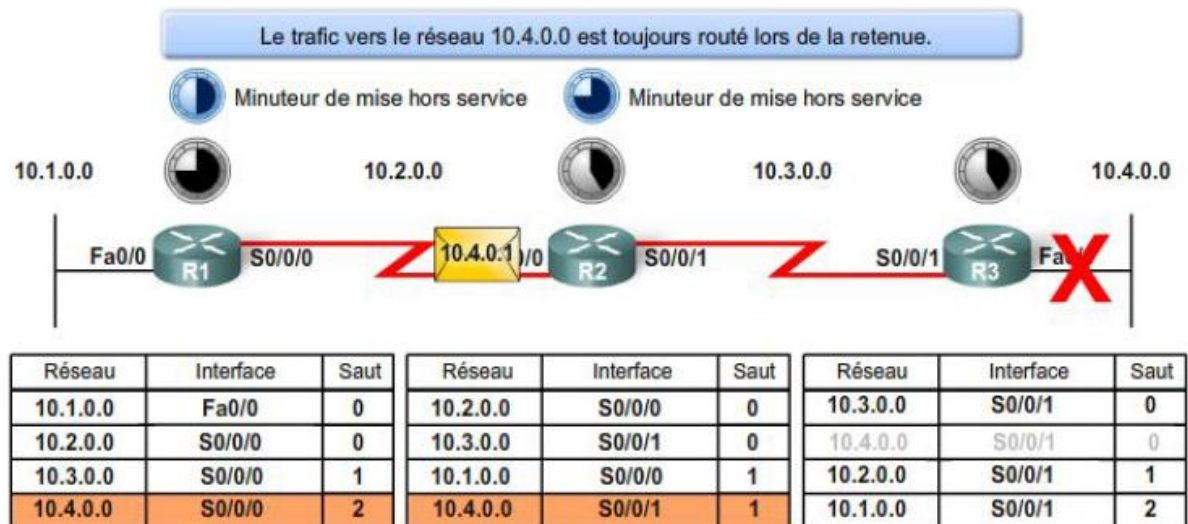
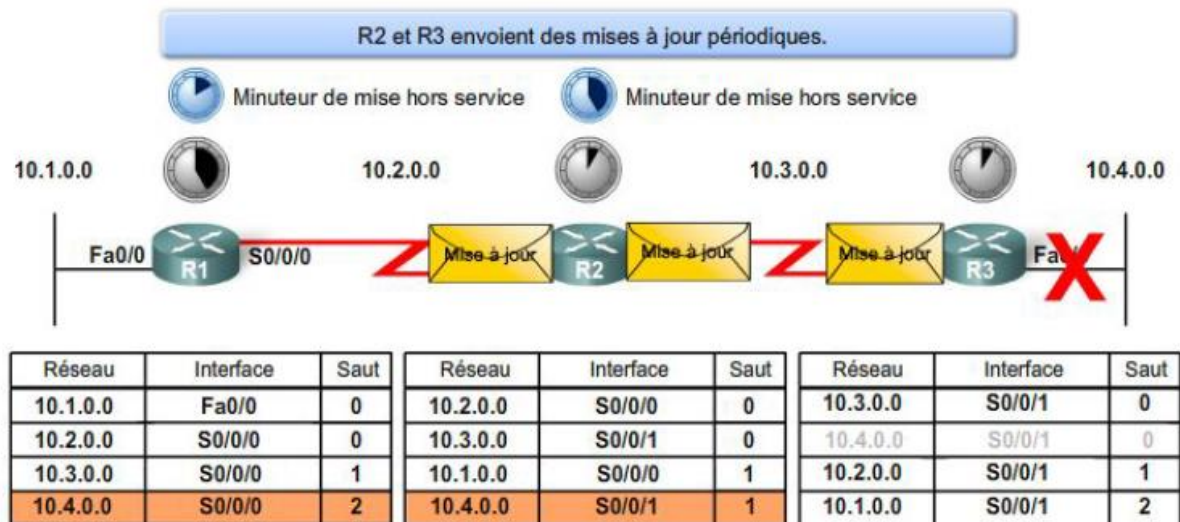
**Si une mise à jour d'un autre voisin est reçue au cours de la période de mise hors service avec une mesure identique ou supérieure pour ce réseau, cette mise à jour est ignorée. Par conséquent, davantage de temps est alloué à la propagation des informations sur la modification.**

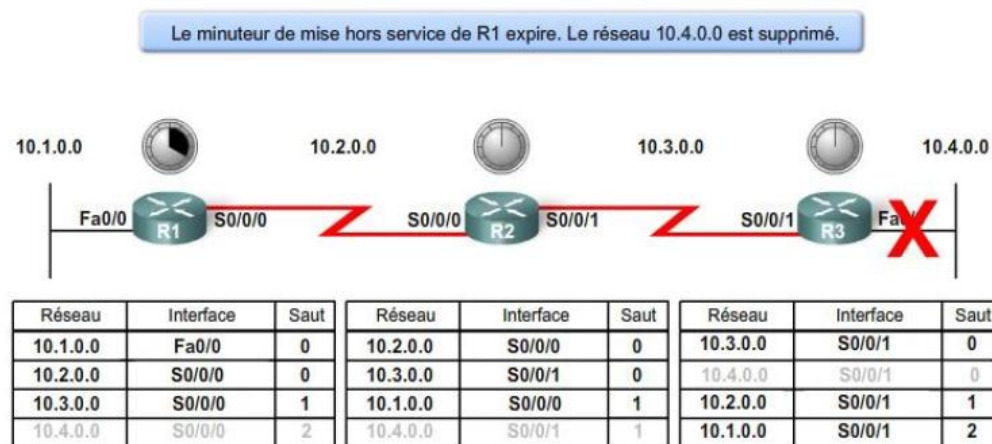
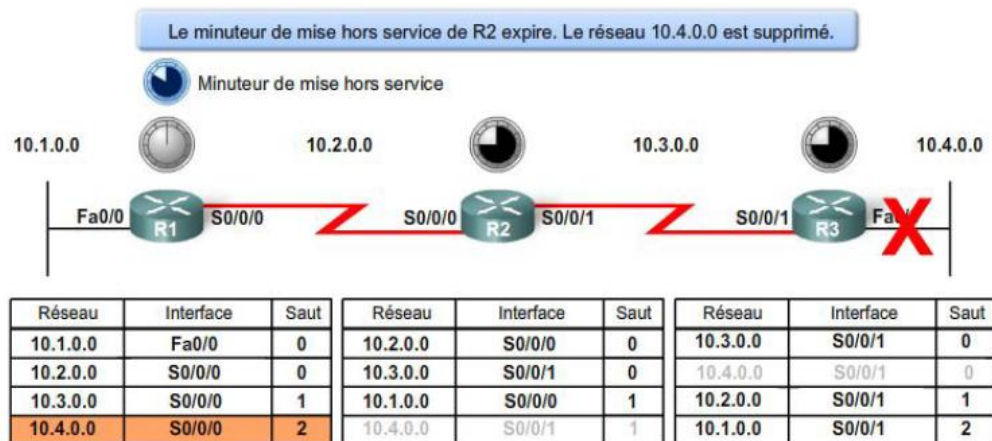
Les routeurs continuent d'acheminer les paquets aux réseaux de destination qui sont marqués comme étant éventuellement inactifs. Le routeur peut ainsi résoudre tout problème lié à une connectivité intermittente. Si le réseau de destination est réellement indisponible et que les paquets sont transférés, un routage de type trou noir est créé et dure jusqu'à l'expiration du minuteur de mise hors service.







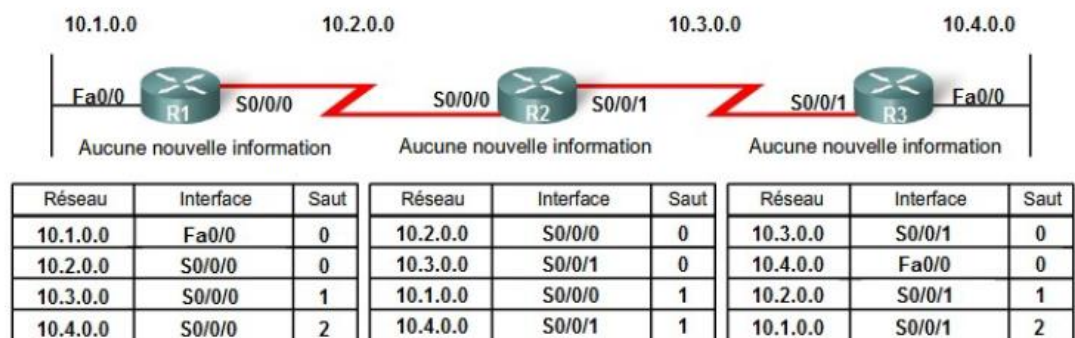




Le réseau est maintenant convergé.

### 3.3.3.2 Découpage d'horizon (Split Horizon)

Le découpage d'horizon est une autre méthode qui permet d'empêcher les boucles de routage provoquées par la convergence lente d'un protocole de routage à vecteur de distance. **Selon la règle de découpage d'horizon, un routeur ne doit pas annoncer de réseau par le biais de l'interface dont est issue la mise à jour.**



R1 n'annonce que le réseau 10.1.0.0 à R2.

R3 n'annonce que le réseau 10.4.0.0 à R2.

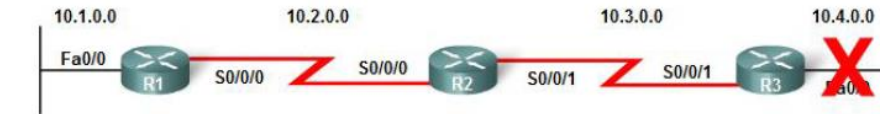
R2 n'annonce que les réseaux 10.3.0.0 et 10.4.0.0 à R1.

R2 n'annonce que les réseaux 10.2.0.0 et 10.1.0.0 à R3.



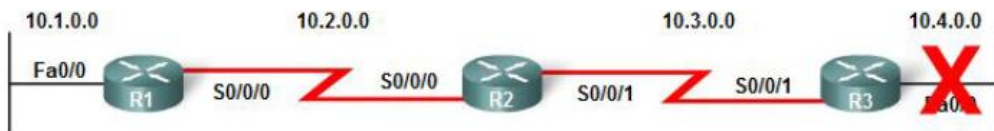
### 3.3.3.3 Empoisonnement de Routing

Le réseau 10.4.0.0 tombe en panne.



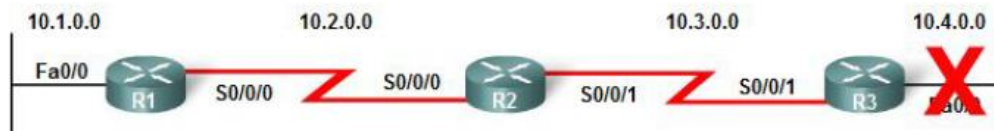
Réseau	Interface	Saut	Réseau	Interface	Saut	Réseau	Interface	Saut
10.1.0.0	Fa0/0	0	10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0
10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0	10.4.0.0	Fa0/0	0
10.3.0.0	S0/0/0	1	10.1.0.0	S0/0/0	1	10.2.0.0	S0/0/1	1
10.4.0.0	S0/0/0	2	10.4.0.0	S0/0/1	1	10.1.0.0	S0/0/1	2

Le réseau 10.4.0.0 tombe en panne.



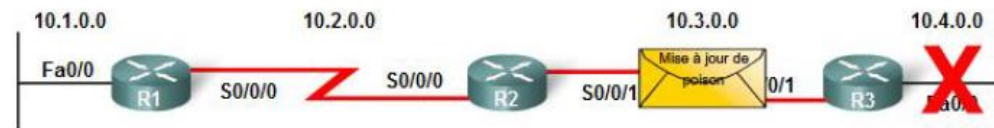
Réseau	Interface	Saut	Réseau	Interface	Saut	Réseau	Interface	Saut
10.1.0.0	Fa0/0	0	10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0
10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0	10.4.0.0	Fa0/0	0
10.3.0.0	S0/0/0	1	10.1.0.0	S0/0/0	1	10.2.0.0	S0/0/1	1
10.4.0.0	S0/0/0	2	10.4.0.0	S0/0/1	1	10.1.0.0	S0/0/1	2

R3 « empoisonne » la route avec une mesure « infinie ».



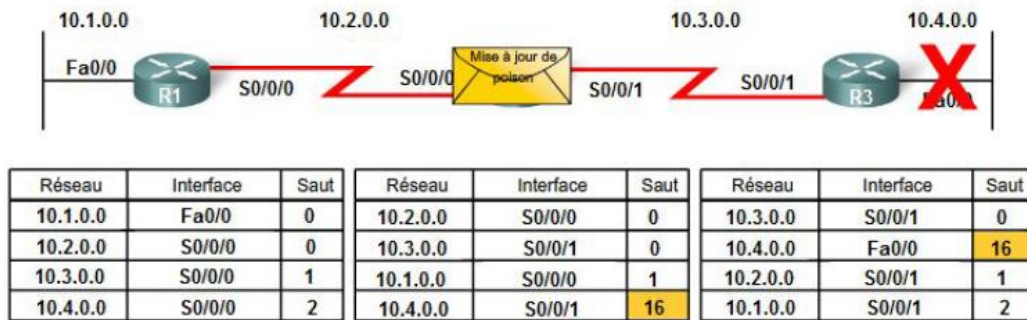
Réseau	Interface	Saut	Réseau	Interface	Saut	Réseau	Interface	Saut
10.1.0.0	Fa0/0	0	10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0
10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0	10.4.0.0	Fa0/0	16
10.3.0.0	S0/0/0	1	10.1.0.0	S0/0/0	1	10.2.0.0	S0/0/1	1
10.4.0.0	S0/0/0	2	10.4.0.0	S0/0/1	1	10.1.0.0	S0/0/1	2

R3 envoie une mise à jour de poison déclenchée à R2.

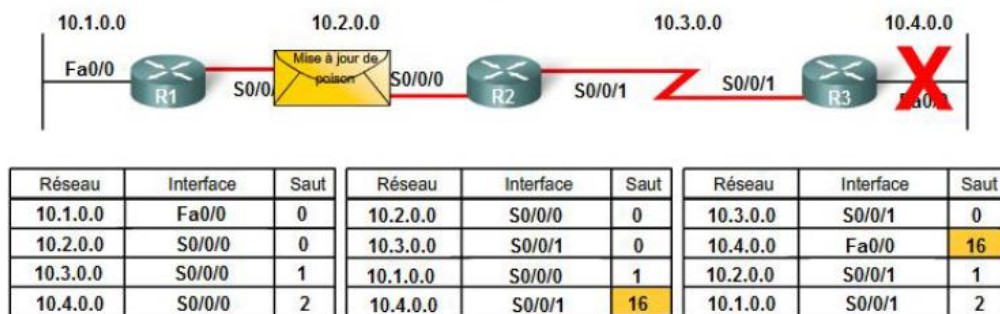


Réseau	Interface	Saut	Réseau	Interface	Saut	Réseau	Interface	Saut
10.1.0.0	Fa0/0	0	10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0
10.2.0.0	S0/0/0	0	10.3.0.0	S0/0/1	0	10.4.0.0	Fa0/0	16
10.3.0.0	S0/0/0	1	10.1.0.0	S0/0/0	1	10.2.0.0	S0/0/1	1
10.4.0.0	S0/0/0	2	10.4.0.0	S0/0/1	1	10.1.0.0	S0/0/1	2

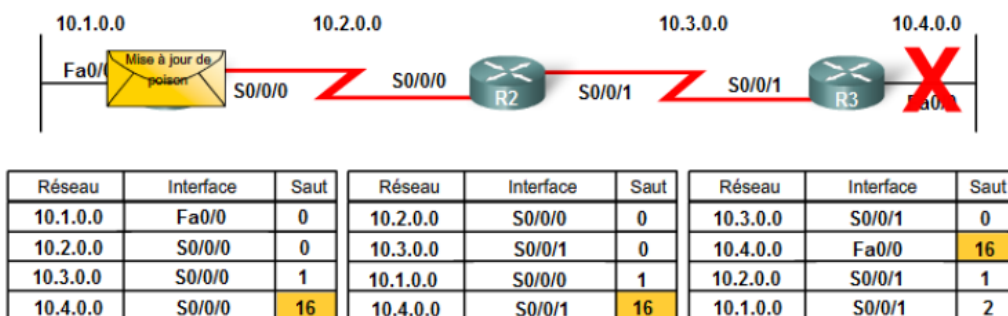
R2 « empoisonne » la route avec une mesure « infinie ».



R2 envoie une mise à jour de poison déclenchée à R1.



R1 « empoisonne » la route avec une mesure « infinie ».



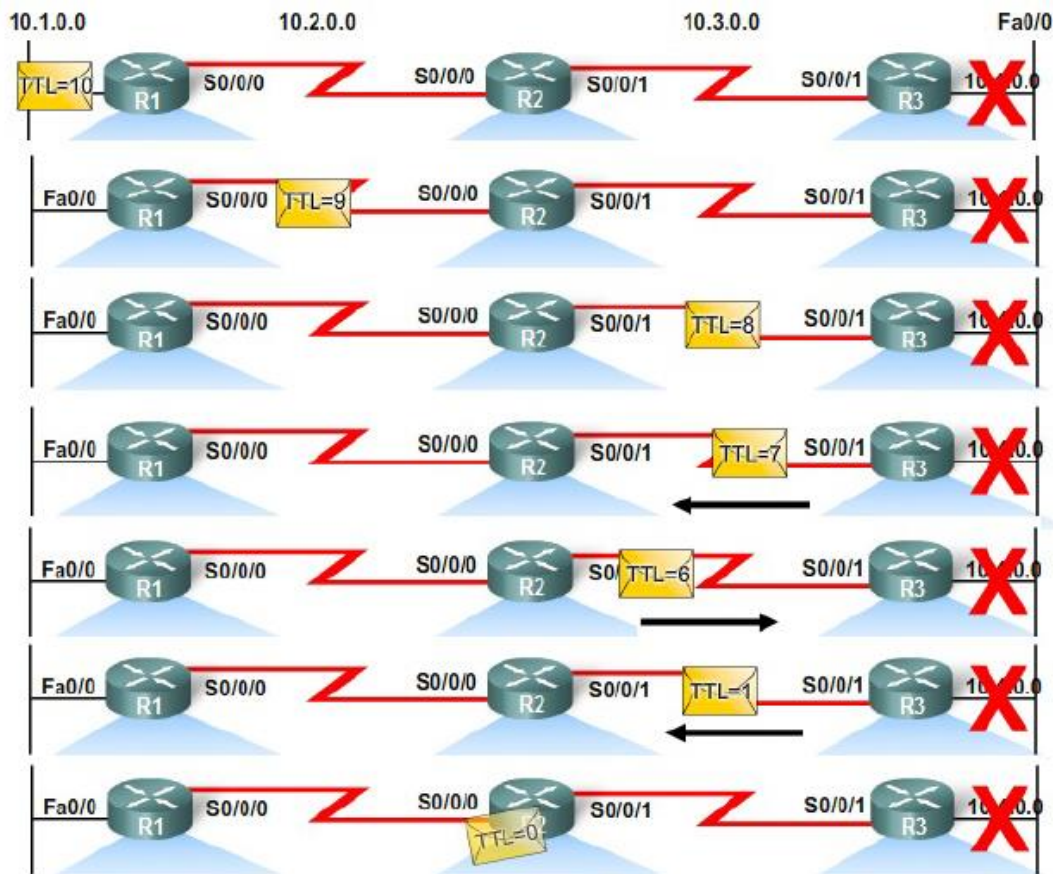
### 3.3.3.4 TTL

Durée de vie est un champ de 8 bits dans l'en-tête IP qui limite le nombre de sauts qu'un paquet peut effectuer à travers le réseau avant d'être supprimé.

Le champ TTL a pour objectif d'éviter la circulation sans fin sur le réseau d'un paquet impossible à remettre. Sa valeur est définie par le périphérique source du paquet.

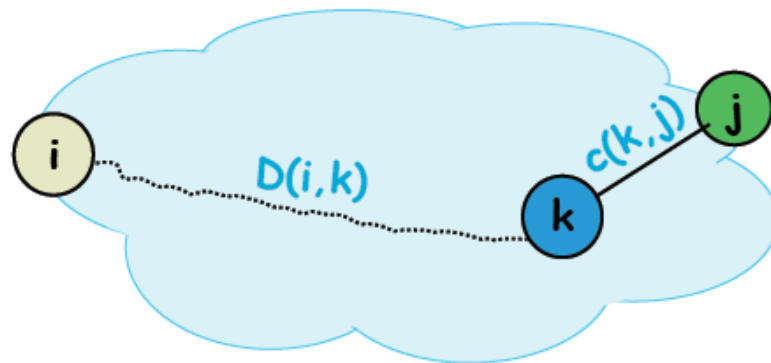
Cette valeur est réduite de 1 par chaque routeur présent sur la route vers sa destination.

Si la valeur du champ TTL atteint zéro avant que le paquet n'arrive à sa destination, ce dernier est supprimé et le routeur renvoie un message d'erreur ICMP (Internet Control Message Protocol) à la source du paquet IP.



#### 4. Approche “état des liens”

- Basé sur la découverte par chaque routeur de la topologie complète du réseau
- Chaque routeur commence par identifier ses voisins
- Chaque routeur transmet ensuite régulièrement à ses voisins :
  - des paquets de mise à jour d’état de lien (Link State Packet) avec :
    - les identifiants de ses voisins
    - les coûts pour les atteindre
  - les paquets LSP qu’il a lui-même reçus
- De proche en proche, chaque routeur :
  - obtient une vision complète de la topologie du réseau
  - peut appliquer un algorithme de type Dijkstra pour calculer les plus courts chemins et déterminer sa table de routage
- Une autre vue de du critère de consistance est utilisée:
  - $D(i,j) = D(i,k) + c(k,j)$

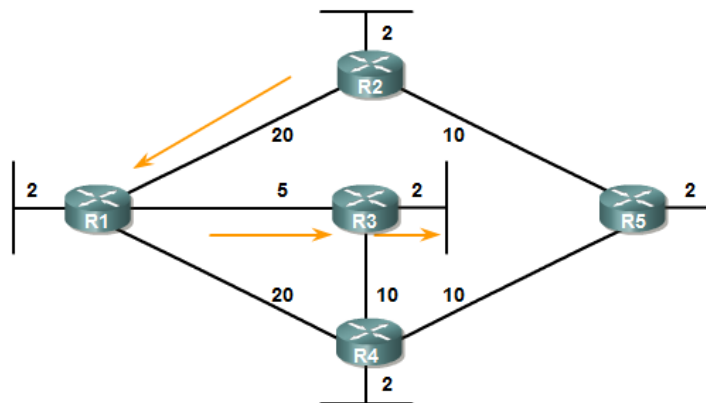


Chaque noeud  $i$  collecte tous les états  $c(*,*)$  d'abord puis exécute localement l'algorithme de plus court chemin (Dijkstra).

L'algorithme de Dijkstra est souvent appelé le shortest path first (SPF)

*Trouver le chemin le plus court (shortest path) est l'objectif de l'algorithme*

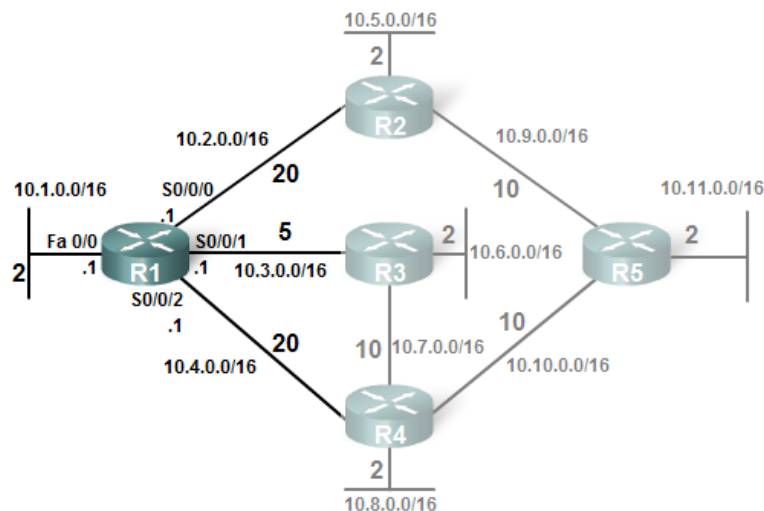
#### 4.1 Etude d'un exemple :



##### Étape 1 : Apprendre sur les réseaux directement connectés

Étape 1 : Chaque routeur apprend sur ses propres liens, grâce aux réseaux directement connectés

- Les interfaces sont configurées avec une adresse et un masque
- Les réseaux directement connectés sont automatiquement inclus dans la table de routage

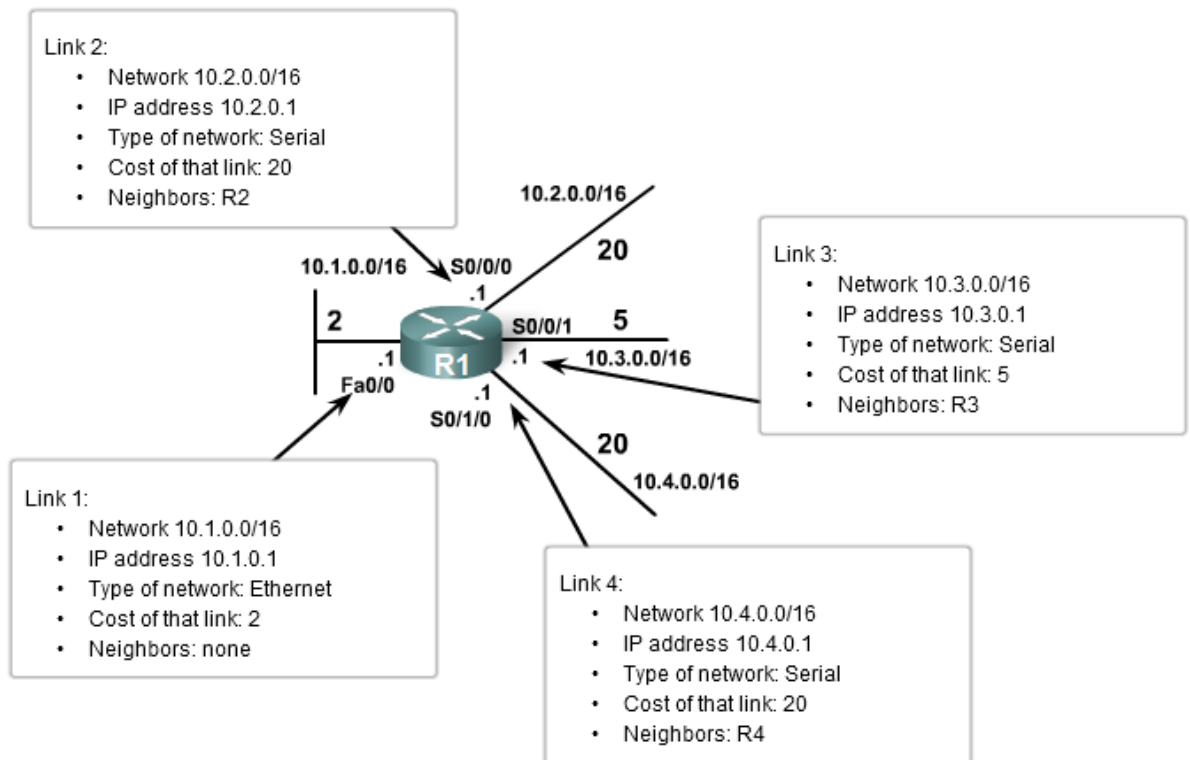




Un lien est une interface dans le routeur

Pour participer au processus de routage, le lien doit :

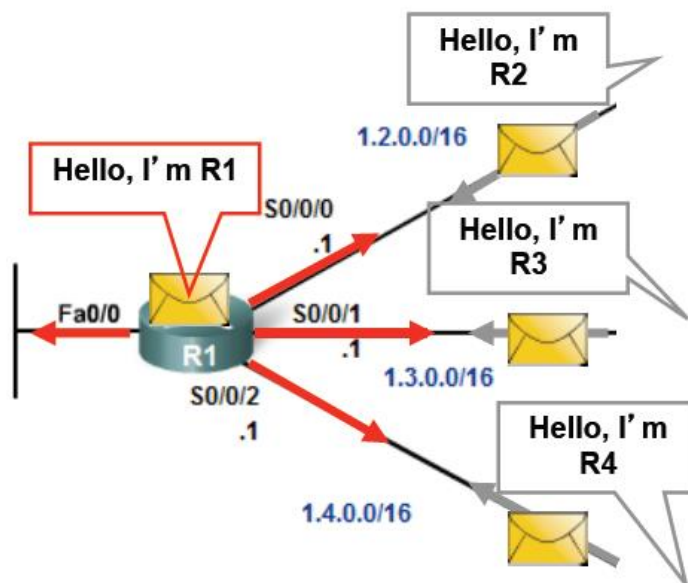
- Être actif (up)
- Être inclus dans l'une des commandes network

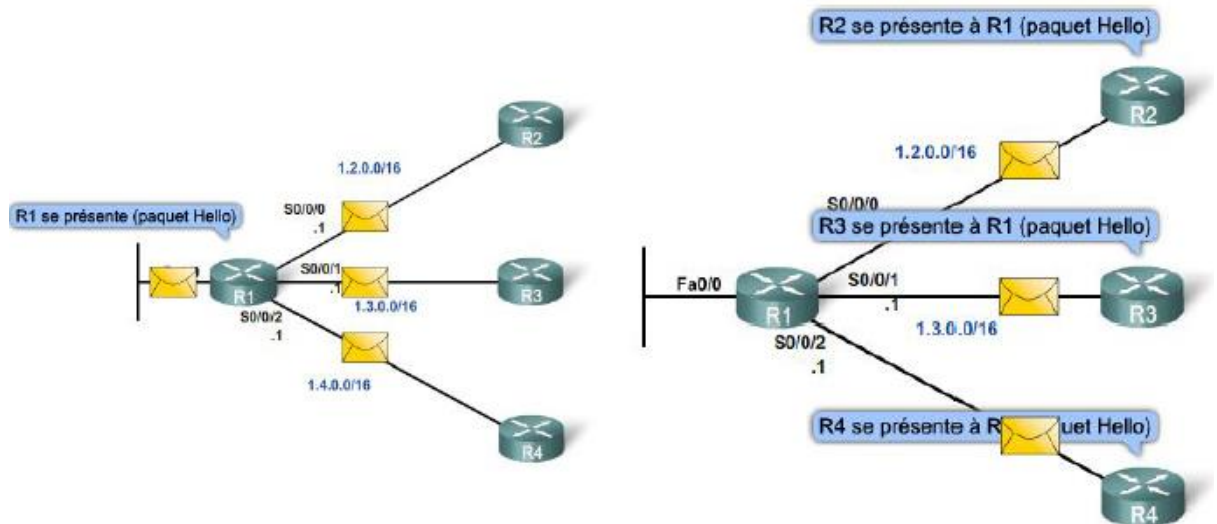


## Étape 2 : Envoyer des Hello aux Voisins

**Chaque routeur est responsable par la découverte des voisins sur un lien**

- Utilisation d'un protocole de Hello pour découvrir les voisins sur un lien
- Un **voisin** est un routeur qui tourne le même type de protocole de routage à état de liens

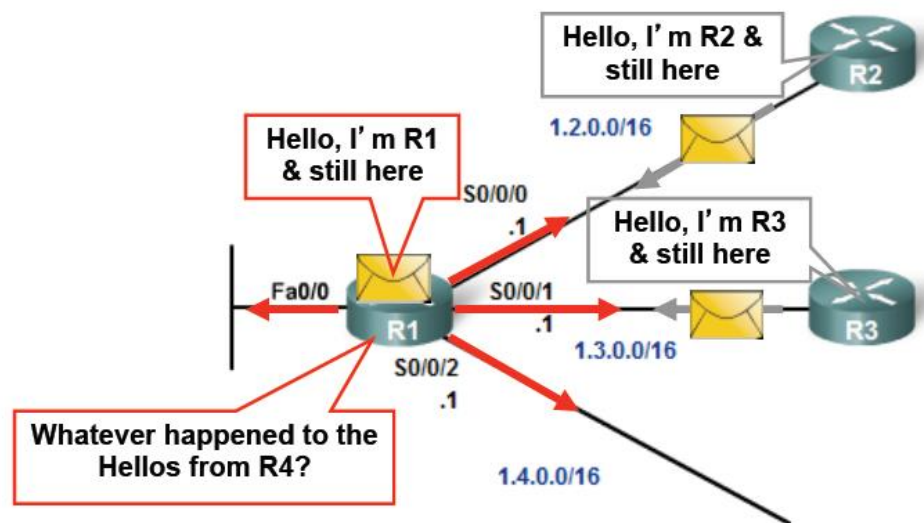




### Paquet Hello

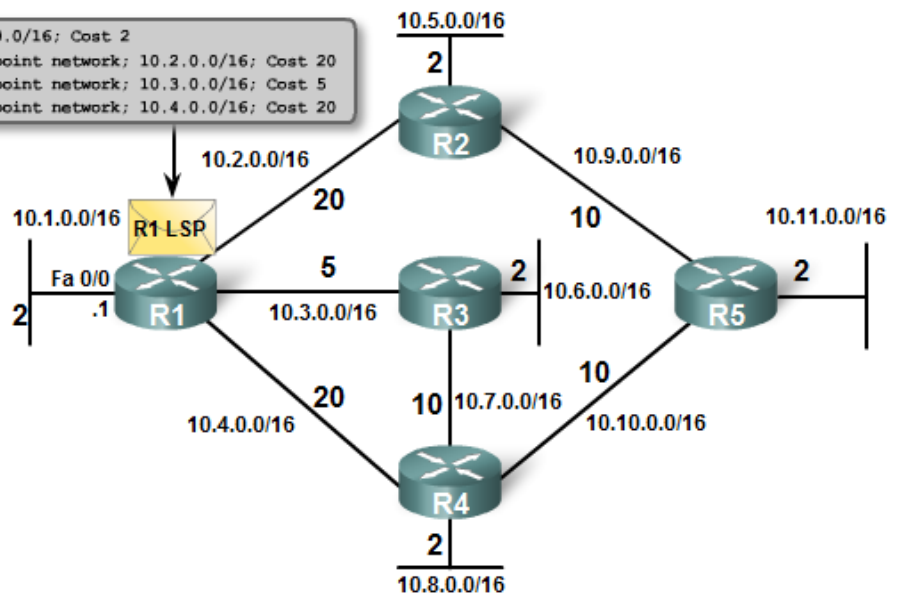
Les paquets Hello ont aussi une fonction "Keepalive"

Si on arrête de recevoir les paquets Hello d'un voisin, ce voisin est considéré comme innatignable et l'adjacence est coupée



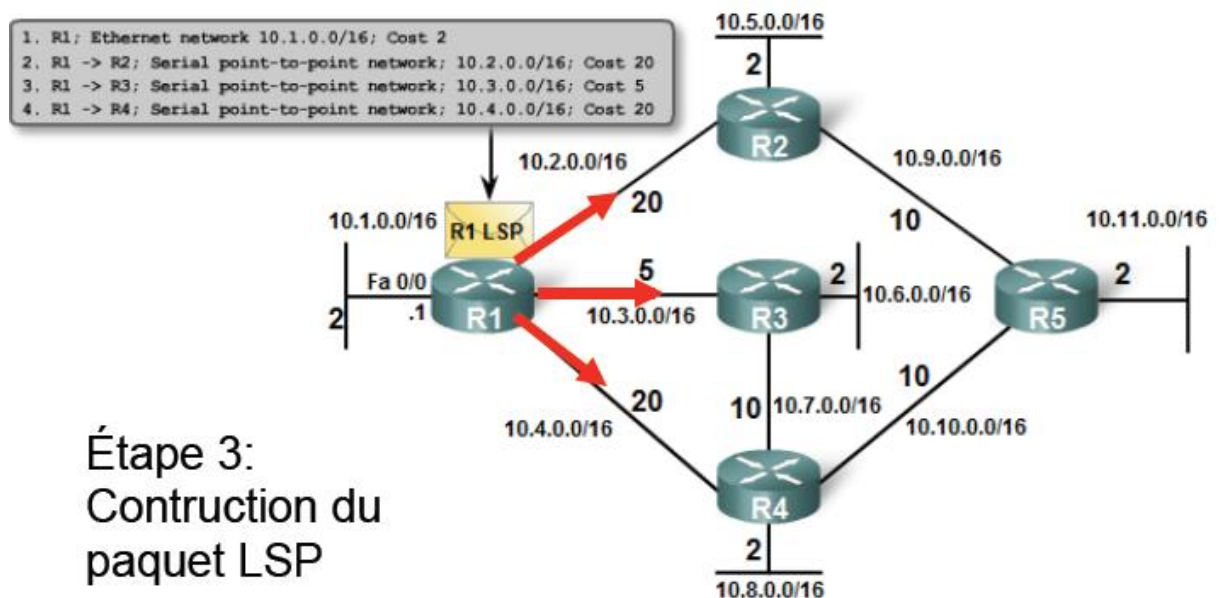
### Étape 3 : Chaque routeur construit son LSP

1. R1; Ethernet network 10.1.0.0/16; Cost 2
2. R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20
3. R1 -> R3; Serial point-to-point network; 10.3.0.0/16; Cost 5
4. R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20



### Après avoir établi les adjacences (voisinage) le routeur

- Construit ses paquets LSP
  - Inclut les informations sur l'état des liens
  - Envoie les LSPs sur toute interface où une adjacence a été établie
- Dans l'exemple, R1 n'envoie pas des LSPs par l'interface Ethernet



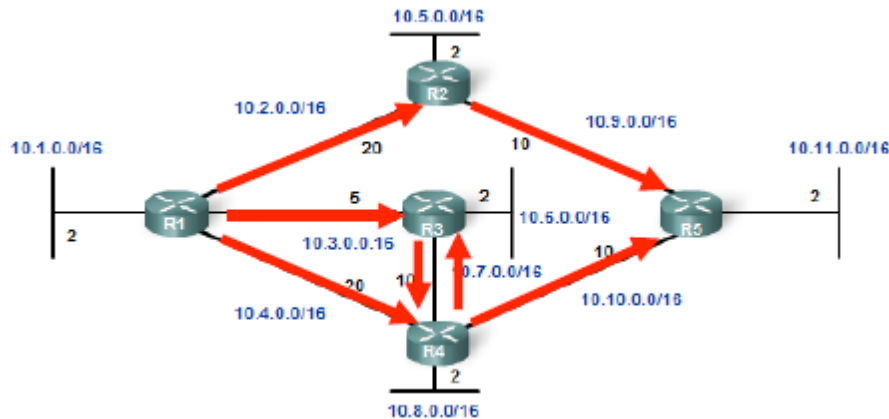
### Étape 4 : Diffusion des paquets LSP aux voisins

Chaque routeur envoie ses LSPs à tous ses voisins, qui les stockent dans une base de données

Chaque routeur inonde ses voisins avec l'information LSP



Lorsqu'un routeur reçoit un paquet LSP d'un voisin, il retransmet ce paquet LSP sur toutes ses interfaces, à l'exception de celle d'où le paquet est venu. Effet de inondation (flooding)



Les LSP ne sont envoyés que :

1. Pendant le démarrage du routeur ou du protocole de routage
2. Toujours où il y a un changement dans la topologie
  - Lien qui tombe
  - Lien activé
  - Nouveau voisin trouvé
  - Déconnexion d'un voisin

### Étape 5 : Construction d'une base de données

Après la propagation des LSPs

Chaque routeur a la totalité des LSPs du réseau

Les LSPs sont stockés dans une base de données d'état des liens

- Étape 5 (dernière étape) :

**Chaque routeur utilise la base de données pour calculer le meilleur chemin vers chaque destination**

- C'est à ce moment que les protocoles à état des liens exécutent l'algorithme SPF

### Base de données d'état des liaisons de R1

#### Paquets LSP de R2 :

- Connecté au voisin R1 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R5 sur le réseau 10.9.0.0/16, coût : 10
- Possède un réseau 10.5.0.0/16, coût : 2

#### Paquets LSP de R3 :

- Connecté au voisin R1 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.7.0.0/16, coût : 10
- Possède un réseau 10.6.0.0/16, coût : 2

#### Paquets LSP de R4 :

- Connecté au voisin R1 sur le réseau 10.4.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.7.0.0/16, coût : 10
- Connecté au voisin R5 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.8.0.0/16, coût : 2

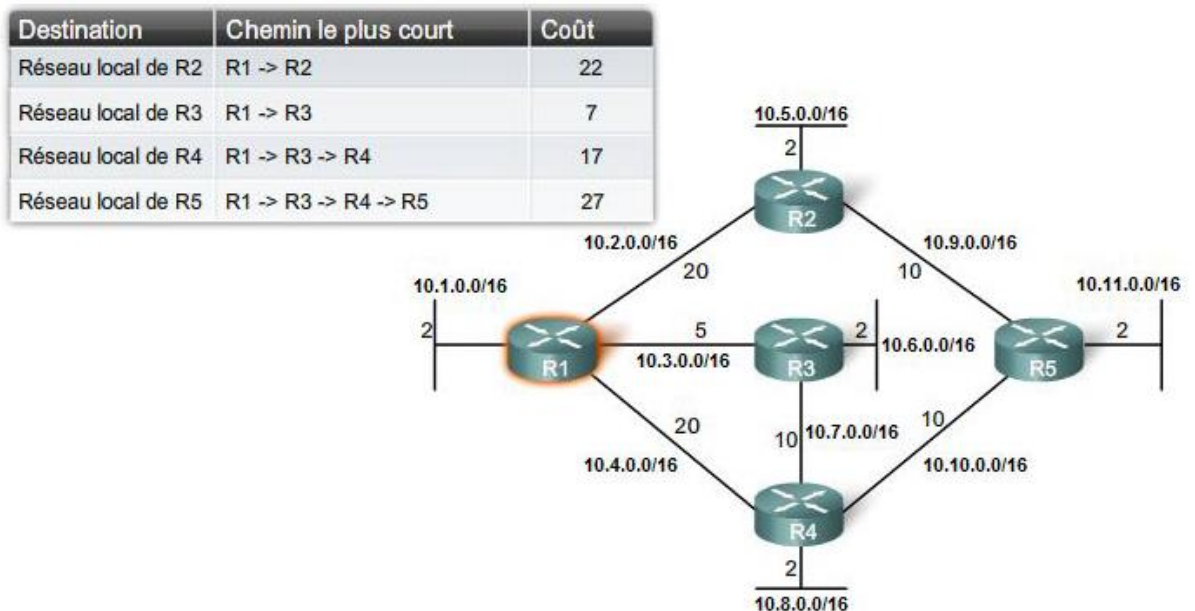
#### Paquets LSP de R5 :

- Connecté au voisin R2 sur le réseau 10.9.0.0/16, coût : 10
- Connecté au voisin R4 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.11.0.0/16, coût : 2

#### États des liaisons de R1 :

- Connecté au voisin R2 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.4.0.0/16, coût : 20
- Possède un réseau 10.1.0.0/16, coût : 2

Avec une base de données complète, R1 peut utiliser l'algorithme SPF pour calculer l'arbre des routes



### Construction de l'arbre SPF

Au début, l'arbre contient seulement les voisins directement connectés

Grâce aux LSPs des autres routeurs, R1 peut maintenant commencer à construire l'arbre SPF (R1 étant la racine)

## Base de données de l'état des liaisons R1

## États des liaisons de R1 :

- Connecté au voisin R2 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.4.0.0/16, coût : 20
- Possède un réseau 10.1.0.0/16, coût : 2

## Paquets LSP de R2 :

- Connecté au voisin R1 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R5 sur le réseau 10.9.0.0/16, coût : 10
- Possède un réseau 10.5.0.0/16, coût : 2

## Paquets LSP de R3 :

- Connecté au voisin R1 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.7.0.0/16, coût : 10
- Possède un réseau 10.6.0.0/16, coût : 2

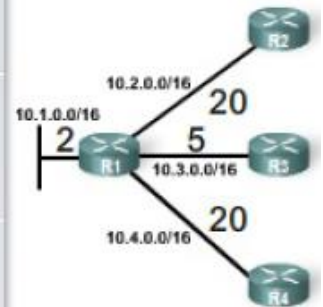
## Paquets LSP de R4 :

- Connecté au voisin R1 sur le réseau 10.4.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.7.0.0/16, coût : 10
- Connecté au voisin R5 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.8.0.0/16, coût : 2

## Paquets LSP de R5 :

- Connecté au voisin R2 sur le réseau 10.9.0.0/16, coût : 10
- Connecté au voisin R4 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.11.0.0/16, coût : 2

## États des liaisons de R1



## Base de données de l'état des liaisons R1

## États des liaisons de R1 :

- Connecté au voisin R2 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.4.0.0/16, coût : 20
- Possède un réseau 10.1.0.0/16, coût : 2

## Paquets LSP de R2 :

- Connecté au voisin R1 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R5 sur le réseau 10.9.0.0/16, coût : 10
- Possède un réseau 10.5.0.0/16, coût : 2

## Paquets LSP de R3 :

- Connecté au voisin R1 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.7.0.0/16, coût : 10
- Possède un réseau 10.6.0.0/16, coût : 2

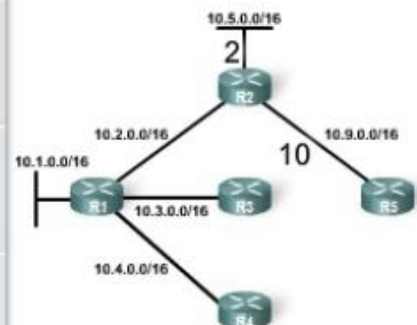
## Paquets LSP de R4 :

- Connecté au voisin R1 sur le réseau 10.4.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.7.0.0/16, coût : 10
- Connecté au voisin R5 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.8.0.0/16, coût : 2

## Paquets LSP de R5 :

- Connecté au voisin R2 sur le réseau 10.9.0.0/16, coût : 10
- Connecté au voisin R4 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.11.0.0/16, coût : 2

## Traitement des paquets LSP de R2



**Base de données de l'état des liaisons R1****États des liaisons de R1 :**

- Connecté au voisin R2 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.4.0.0/16, coût : 20
- Possède un réseau 10.1.0.0/16, coût : 2

**Paquets LSP de R2 :**

- Connecté au voisin R1 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R5 sur le réseau 10.9.0.0/16, coût : 10
- Possède un réseau 10.5.0.0/16, coût : 2

**Paquets LSP de R3 :**

- Connecté au voisin R1 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.7.0.0/16, coût : 10
- Possède un réseau 10.6.0.0/16, coût : 2

**Paquets LSP de R4 :**

- Connecté au voisin R1 sur le réseau 10.4.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.7.0.0/16, coût : 10
- Connecté au voisin R5 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.8.0.0/16, coût : 2

**Paquets LSP de R5 :**

- Connecté au voisin R2 sur le réseau 10.9.0.0/16, coût : 10
- Connecté au voisin R4 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.11.0.0/16, coût : 2

**Traitement des paquets LSP de R3****Base de données de l'état des liaisons R1****États des liaisons de R1 :**

- Connecté au voisin R2 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.4.0.0/16, coût : 20
- Possède un réseau 10.1.0.0/16, coût : 2

**Paquets LSP de R2 :**

- Connecté au voisin R1 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R5 sur le réseau 10.9.0.0/16, coût : 10
- Possède un réseau 10.5.0.0/16, coût : 2

**Paquets LSP de R3 :**

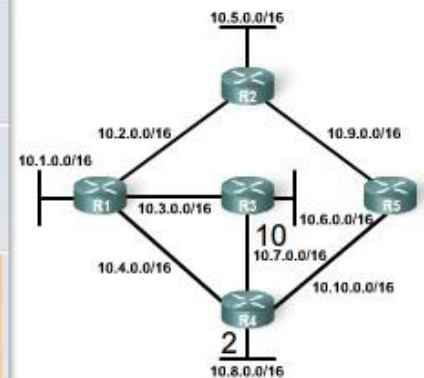
- Connecté au voisin R1 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.7.0.0/16, coût : 10
- Possède un réseau 10.6.0.0/16, coût : 2

**Paquets LSP de R4 :**

- Connecté au voisin R1 sur le réseau 10.4.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.7.0.0/16, coût : 10
- Connecté au voisin R5 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.8.0.0/16, coût : 2

**Paquets LSP de R5 :**

- Connecté au voisin R2 sur le réseau 10.9.0.0/16, coût : 10
- Connecté au voisin R4 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.11.0.0/16, coût : 2

**Traitement des paquets LSP de R4**



**Base de données de l'état des liaisons R1**

## États des liaisons de R1 :

- Connecté au voisin R2 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.4.0.0/16, coût : 20
- Possède un réseau 10.1.0.0/16, coût : 2

## Paquets LSP de R2 :

- Connecté au voisin R1 sur le réseau 10.2.0.0/16, coût : 20
- Connecté au voisin R5 sur le réseau 10.9.0.0/16, coût : 10
- Possède un réseau 10.5.0.0/16, coût : 2

## Paquets LSP de R3 :

- Connecté au voisin R1 sur le réseau 10.3.0.0/16, coût : 5
- Connecté au voisin R4 sur le réseau 10.7.0.0/16, coût : 10
- Possède un réseau 10.6.0.0/16, coût : 2

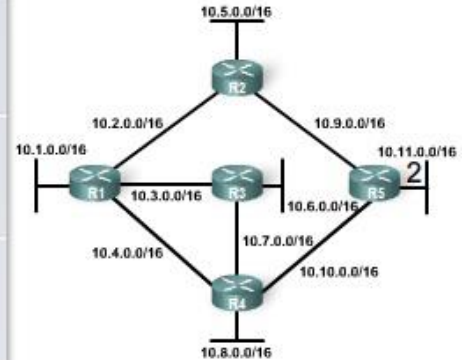
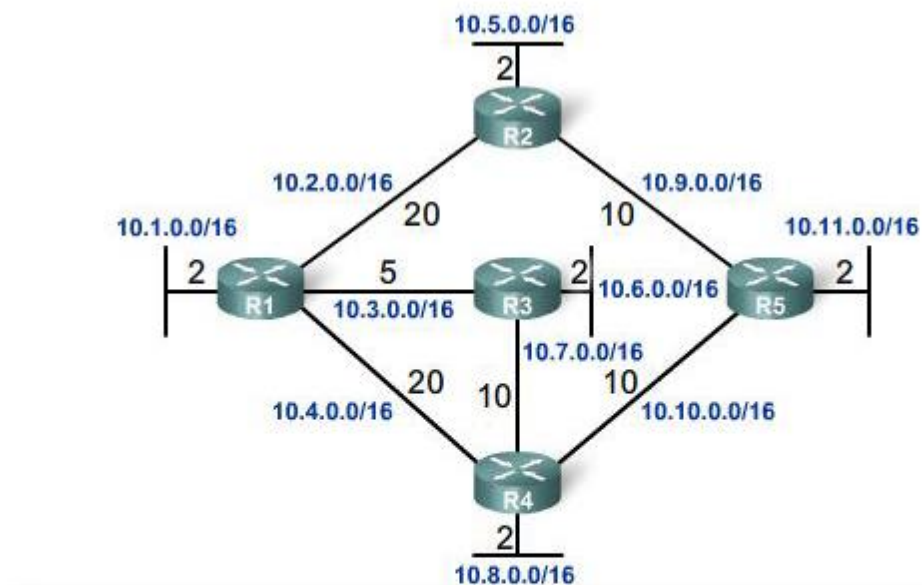
## Paquets LSP de R4 :

- Connecté au voisin R1 sur le réseau 10.4.0.0/16, coût : 20
- Connecté au voisin R3 sur le réseau 10.7.0.0/16, coût : 10
- Connecté au voisin R5 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.8.0.0/16, coût : 2

## Paquets LSP de R5 :

- Connecté au voisin R2 sur le réseau 10.9.0.0/16, coût : 10
- Connecté au voisin R4 sur le réseau 10.10.0.0/16, coût : 10
- Possède un réseau 10.11.0.0/16, coût : 2

## Traitement des paquets LSP de R5

**SPF R1 final pour Les LAN**

Destination	Chemin le plus court	Coût
Réseau local R2	R1 vers R2	22
Réseau local R3	R1 vers R3	7
Réseau local R4	R1 vers R3 vers R4	17
Réseau local R5	R1 vers R3 vers R4 vers R5	27

R1 détermine que le chemin le plus court vers chaque réseau distant :

- le réseau 10.5.0.0/16 via R2 série 0/0/0, pour un coût de 22
- le réseau 10.6.0.0/16 via R3 série 0/0/1 pour un coût de 7
- le réseau 10.7.0.0/16 via R3 série 0/0/1 pour un coût de 15
- le réseau 10.8.0.0/16 via R3 série 0/0/1 pour un coût de 17

- le réseau 10.9.0.0/16 via R2 série 0/0/0 pour un coût de 30
- le réseau 10.10.0.0/16 via R3 série 0/0/1 pour un coût de 25
- le réseau 10.11.0.0/16 via R3 série 0/0/1 pour un coût de 27

### Remarque :

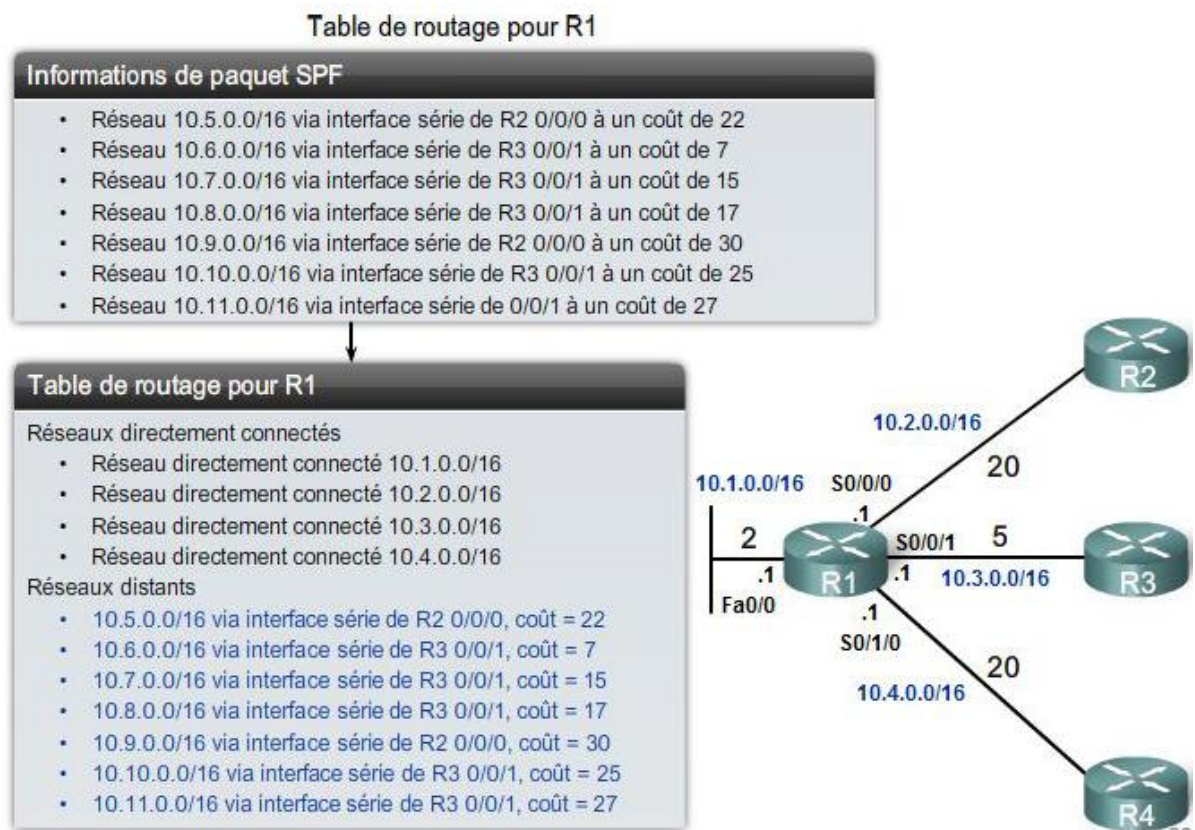
Chaque routeur construit son arbre SPF indépendamment des autres routeurs  
Pour un bon fonctionnement, les bases de données LSP doivent être identiques partout !

### Ajout des routes dans la table de routage

Les chemins établis peuvent être insérés dans la table de routage

La table de routage indiquera aussi

- Les réseaux directement connectés
- Les routes des autres sources telles que les routes statiques
- Les paquets peuvent être transférés selon ces entrées dans la table de routage



### Avantage des protocoles de routage à état de liaison

- Élaboration d'une carte topologique : chaque routeur crée sa propre carte topologique du réseau pour déterminer le chemin le plus court
- l'inondation immédiate de paquets LSP permet une convergence plus rapide
- les paquets LSP ne sont envoyés qu'en cas de modification de la topologie et contiennent uniquement les informations concernant cette modification une conception hiérarchique est utilisée lors de l'implémentation de plusieurs zones.

## Contenu

1. Protocoles de routage .....	1
1.1 Classification des algorithmes de routage .....	1
1.2 AS (Autonomous System) .....	2
1.3 IGP (Interior Gateway Protocol) .....	2
1.4 EGP (Exterior Gateway Protocol) .....	2
2. Routage: principes de base .....	2
2.1 Critère de “consistance” .....	3
2.2 Notion de convergence .....	3
3. L’approche vecteur de distance .....	4
3.1 Vecteur de distance (Distance-Vector, DV) .....	4
3.2 Algorithme DV .....	5
3.3 Problèmes avec le Distance Vector .....	6
3.3.1 Boucle de routage .....	6
3.3.2 Le comptage à l’infini .....	9
3.3.3 Solutions .....	11
4. Approche “état des liens” .....	18
4.1 Etude d’un exemple : .....	19