



# Filière Sciences Mathématiques et Informatiques (SMI-S6) (Cours Réseaux II-**suite**)

K. Ibrahimi, FSK

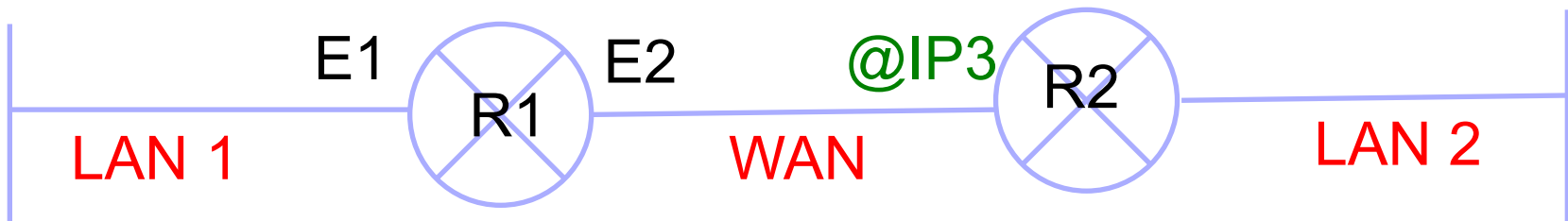
# Routage des paquets IP

- La table de routage est un regroupement d'informations permettant de déterminer le prochain routeur à utiliser pour accéder à un réseau précis sur lequel se trouvera la machine avec laquelle nous souhaitons dialoguer.
- Les informations de de la table de routage :
  - Adresse IP destinataire (réseau distant)
  - Masque du réseau ou du sous réseau
  - Passerelle (adresse IP du prochain saut) ou bien le nom de l'interface physique de sortie du routeur

# Identification de la route

- Masque réseau permet
  - d'identifier les routes potentielles dans la table de routage
  - Vérifier si l'adresse IP d'un hôte s'elle est locale ou distante
- Passerelle
  - La passerelle est l'adresse IP du prochain routeur directement connecté vers lequel un paquet IP doit être envoyé
- Choix de la route par le routeur
  - La route est choisie en fonction de maximum de vraisemblance sur les octets les plus semblables entre @IP destination et @IP extraite:  
$$@IP_{extraite} = @IP_{destination} \text{ ET } @IP_{masque}$$

# Structure de la table de routage IP



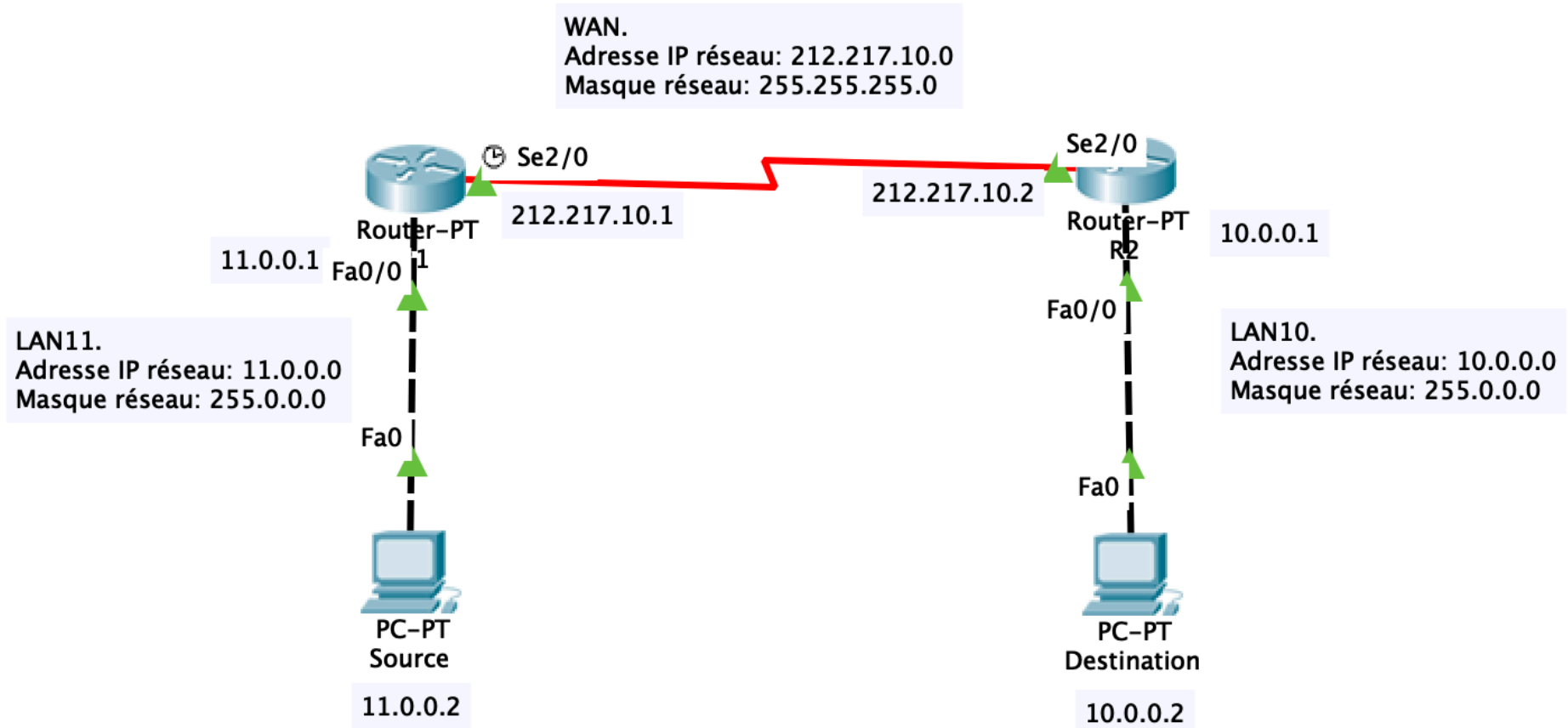
## Routeur R1

Type	@IP du réseau/sous réseau distant	@IP du masque réseau/sous-réseau	Interface physique de sortie du routeur	@IP Passerelle / prochain saut	Métrieque
Réseau directement connecté	LAN 1	Masque 1	E1		0
Réseau directement connecté	WAN	Masque 2	E2		0
Réseau distant	LAN 2	Masque 3	E2	@IP3	1
Route par défaut	0.0.0.0	0.0 .0.0	E2	@IP3	1

# Types de routage des paquets IP

- **Statique** : La table de routage est entrée manuellement par l'administrateur du réseau informatique.
- **Dynamique** : Le routeur construit lui-même sa table de routage en fonction des informations qu'il reçoit des protocoles de routage des autres routeurs.
- Le routage a deux fonctions principales :
  - **Mis à jour des tables de routage**
  - **Acheminement des paquets selon des critères:**

# Routage statique (exemple)



# Routage statique (configuration)

## Etapes de la configuration :

**Etape 1:** Configuration de base du routeur

- Passer en mode privilège du routeur (#)
- Nommer votre routeur
- Configurer les interfaces LAN (Ethernet, FastEthernet, GigaEthernet, ....)
- Configurer les interfaces WAN (serial,...)

**Etape 2:** Création des routes par l'administrateur vers les réseaux distants

**Etape 3:** Vérification de la configuration

**Etape 4:** Sauvegarde de la configuration

# Routage statique (configuration)

```
Router>enable
```

```
Router(config)#hostname R1
```

```
Interface LAN11
```

```
R1(config)#interface fastEthernet 0/0
```

```
R1(config-if)#ip address 11.0.0.1 255.0.0.0
```

```
R1(config-if)#no shutdown
```

```
R1(config-if)#exit
```

```
Interface WAN
```

```
R1(config)#interface serial 2/0
```

```
R1(config-if)#ip address 212.217.10.1 255.255.255.0
```

```
R1(config-if)#clock rate 64000
```

```
R1(config-if)#no shutdown
```

```
Route statique vers le réseau LAN10
```

```
R1(config)#ip route 10.0.0.0 255.0.0.0 212.217.10.2
```

```
R1(config)#ip route 0.0.0.0 0.0.0.0 212.217.10.2
```

```
R1(config)#end
```

```
R1#wr
```



# Routage statique (configuration)

```
Router>enable
```

```
Router(config)#hostname R2
```

```
Interface LAN10
```

```
R2(config)#interface fastEthernet 0/0
```

```
R2(config-if)#ip address 10.0.0.1 255.0.0.0
```

```
R2(config-if)#no shutdown
```

```
R2(config-if)#exit
```

```
Interface WAN
```

```
R2(config)#interface serial 2/0
```

```
R2(config-if)#ip address 212.217.10.2 255.255.255.0
```

```
R2(config-if)#no shutdown
```

```
Route statique vers le réseau LAN11
```

```
R2(config)#ip route 11.0.0.0 255.0.0.0 212.217.10.1
```

```
R2(config)#ip route 0.0.0.0 0.0.0.0 212.217.10.1
```

```
R2(config)#end
```

```
R2#wr
```

# Routage statique (configuration)

Table de routage de R1:

R1#show ip route

Codes: C - connected, S - static, \* - candidate default

S 10.0.0.0/8 [1/0] via 212.217.10.2

C 11.0.0.0/8 is directly connected, FastEthernet0/0

C 212.217.10.0/24 is directly connected, Serial2/0

S\* 0.0.0.0/0 [1/0] via 212.217.10.2

Table de routage de R2:

R2#show ip route

Codes: C - connected, S - static, \* - candidate default

C 10.0.0.0/8 is directly connected, FastEthernet0/0

S 11.0.0.0/8 [1/0] via 212.217.10.1

C 212.217.10.0/24 is directly connected, Serial2/0

S\* 0.0.0.0/0 [1/0] via 212.217.10.1

# Routage statique ou dynamique

- **Routage statique**: si une voie entre deux routeurs distants est hors fonction, le routeur émetteur ne sera pas en mesure de transmettre le paquet à un autre routeur via cette voie.
- **Le routage dynamique** offre une plus grande souplesse automatique. Lorsque le routeur émetteur reconnaît que le lien vers un routeur est hors fonction, il modifie sa table de routage, faisant de la voie qui passe par un autre routeur sa voie privilégiée pour atteindre le réseau destinataire.

# Classes de protocole de routage

## ❖ Vecteur de distance (distance vector - DV)

- Algorithme par diffusion de la table de routage
- Routage distribué basé sur l'algorithme de Bellman-ford
- Entre voisins directs, périodiquement
- Métrique : nombre de sauts
- Exemple de protocole: RIPv1, RIPv2, IGRP

## ❖ Etat de liens (Link state - LS)

- Diffusion à tous de la connaissance sur les liaisons locales du routeur
- Calcul local des meilleurs chemins (Short Path First-SPF)
- Routage décentralisé basé sur l'algorithme de Dijkstra
- Exemple de protocole: OSPF, IS-IS

# Routage par classe et sans classe

- Routeur fonctionne par classe/**classfull**, s'il est associé à une adresse de classe A, B ou C. **Ex. RIPv1, IGRP.** Ces protocoles n'envoient pas des masques sous réseaux dans les messages de mise à jour de routage.
- Routeur fonctionne sans classe/**classless**, s'il fonctionne avec sous réseau. **Ex. RIPv2, EIGRP, OSPF, IS-IS.** Ces protocoles envoient des masques sous réseaux dans les messages de mise à jour de routage.
- Les mises à jour prennent un certain temps avant d'atteindre tous les routeurs.
- Pendant le temps de mise à jour, des routes nouvelles peuvent apparaître et d'autres disparaître.
- Ce temps est le temps de convergence.
- Au démarrage du routeur, que les routes vers les réseaux directement connectés sont connait.
- Chaque routeur prend sa décision seul en se basant sur sa table de routage.
- Le routage se fait de proche en proche.
- **Une route ajoutée dans un sens, elle doit être ajoutée dans l'autre sens.**
- Le protocole RIP résume automatiquement les mises à jour entre les réseaux par classe.



# **Routage à vecteur de distance (DV)**

# Routage à Vecteur de distance (DV)

- Chaque routeur dispose d'une table de routage précisant pour chaque réseau destinataire:  
la meilleure distance connue et la ligne par ou l'atteindre; c'est le « vecteur de distance »
- Chaque routeur échange périodiquement ses informations avec ses voisins et met à jour ses tables de routage.
- La métrique peut être en sauts,
- L' algorithme Bellman-Ford est à vecteur de distance.

# Routage distribué de Bellman-Ford

- **Itératif.** Chaque itération est suite:
  - Changement du coût d'un lien local
  - Réception d'un message d'un voisin indiquant le changement d'un chemin du coût minimal
  - **Self-terminating**: quand pas d'échange d'information, l'algorithme arrête.
- **Distribué :**
  - Aucun routeur n'a d'information globale.
  - Chaque routeur communique avec ses voisins uniquement.
- **Table de distance:**
  - Chaque routeur a sa table de distance pour atteindre les autres nœuds.

$$D^X(Y,Z) = \begin{array}{l} \text{distance de } X \text{ vers } Y, \\ \text{via } Z \text{ comme prochain saut} \end{array}$$

$$D^X(Y,Z) = c(X,Z) + \min_w \{D^Z(Y,w)\}$$



# Algorithme de mis-à-jour distribué

- Etat initial, chaque routeur connaît :
  - Son adresse
  - Ses interfaces
- Un message de routage (mdr) contient une liste de couples:
  - (@IPréseau1\_distant, distance1)
  - (@IPréseau2\_distant, distance2)
  - (@IPréseau3\_distant, distance3)
  - (@IPréseau4\_distant, distance4)
  - .....
- Lorsqu'un routeur reçoit un mdr, **chaque couple** est comparé aux entrées de la table de routage.

# Algorithme de mis-à-jour

## Cas possibles du message de routage (mdr):

1. Le couple n'existe pas dans la table de routage et la métrique est finie :
  - *ajout d'une nouvelle ligne dans la table de routage* :
    - prochain routeur = routeur de provenance du mdr;
    - distance = distance reçue + 1
2. Le couple existe et sa métrique est supérieure à celle reçue :
  - *mis-à-jour de l'entrée* :
    - distance = distance reçue + 1
3. Le couple existe et son prochain routeur est celui d'où provient la liste (**changement de la distance**)
  - *mis-à-jour de l'entrée* :
    - distance = distance reçue + 1
4. Sinon rien.

# Algorithme de mis-à-jour

## Cas 1:

$\text{mdr} = (\text{réseau (resR1)}, \text{distance reçu (dr)})$

Si  $\text{dr} < \infty$  alors :

- ajout d'une nouvelle ligne dans la table de routage :
  - prochain routeur = routeur de provenance du mdr;
  - distance = distance reçue + 1

Réseau distant	Prochain routeur	Distance
resR1	Routeur d'envoi du mdr	dr+1

$\infty$ : distance infinie signifie que le réseau distant est injoignable.

# Algorithme de mis-à-jour

## Cas 2:

$\text{mdr} = (\text{réseau (resR2)}, \text{distance reçu (dr)})$

Si le couple existe et sa métrique est supérieure à celle reçue :

- *mis-à-jour de l'entrée* :
  - $\text{distance} = \text{distance reçue} + 1$

Réseau distant	Prochain routeur	Distance
resR2	Routeur d'envoi du mdr	dr+1

# Algorithme de mis-à-jour

## Cas 3:

$\text{mdr} = (\text{réseau (resR3)}, \text{distance reçu (dr)})$

Le couple existe et son prochain routeur est celui d'où provient la liste (**changement de la distance**)

- *mis-à-jour de l'entrée :*
  - **distance=distance reçue + 1**

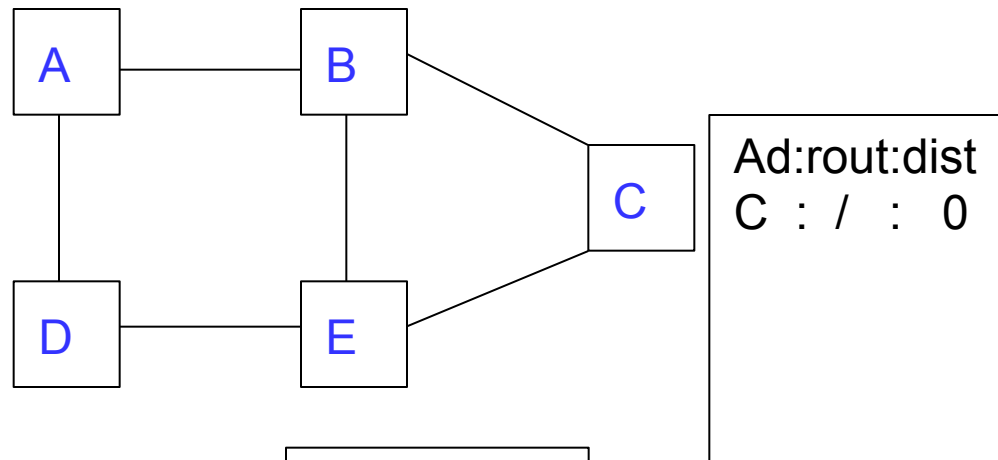
Réseau distant	Prochain routeur	Distance
resR3	Routeur d'envoi du mdr	dr+1

# Exemple

Trouver la table de routage/distance de chaque nœud?

Ad:rout:dist  
A : / : 0

Ad:rout:dist  
B : / : 0



Ad:rout:dist  
C : / : 0

Ad:rout:dist  
D : / : 0

Ad:rout:dist  
E : / : 0

# Exemple

Ad:rout:dist  
A : / : 0

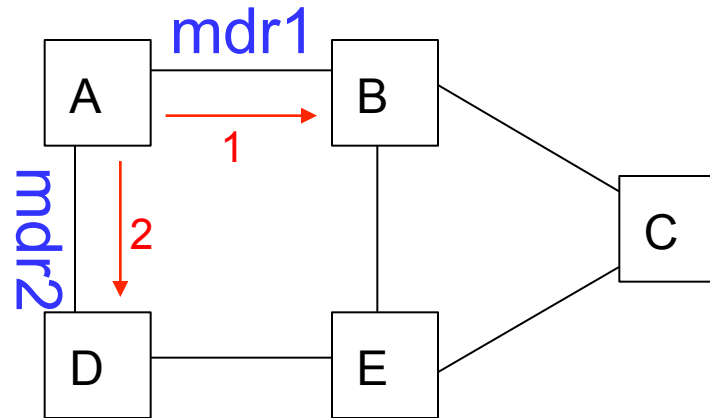
Ad:rout:dist  
B : / : 0

Noeud A envoie sa table de routage à ses voisins:

mdr 1 = A->B; A: 0

mdr 2 = A->D; A: 0

Ad: distant/noeud  
rout: via  
dist: distance

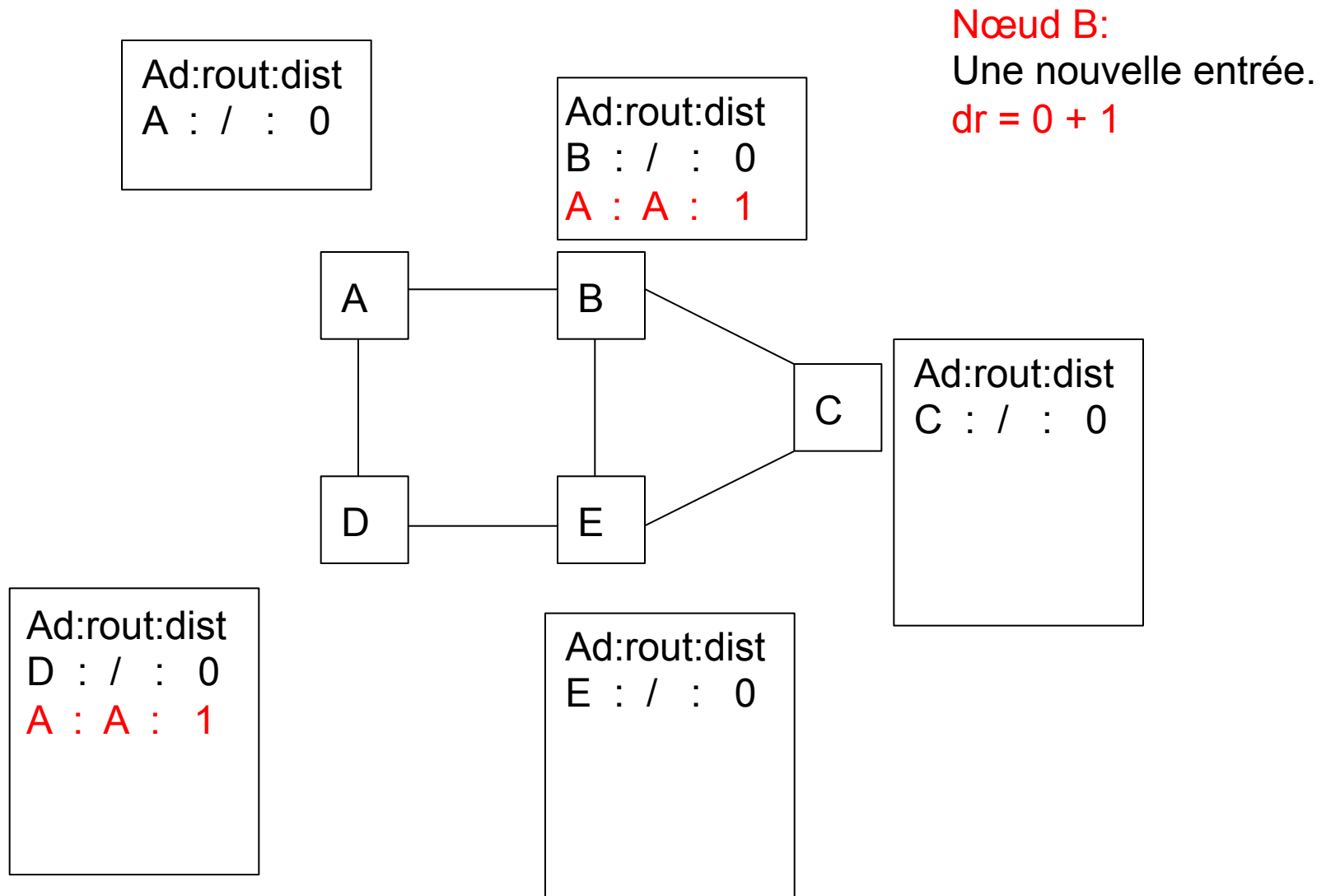


Ad:rout:dist  
C : / : 0

Ad:rout:dist  
D : / : 0

Ad:rout:dist  
E : / : 0

# Exemple





# Exemple

Ad:rout:dist  
A : / : 0

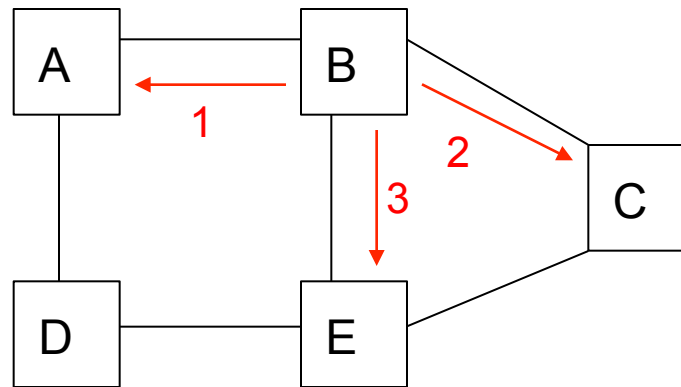
Ad:rout:dist  
B : / : 0  
A : A : 1

Nœud B:

mdr 1 : B->A;B:0;A:1

mdr 2 : B->C;B:0;A:1

mdr 3 : B->E;B:0;A:1

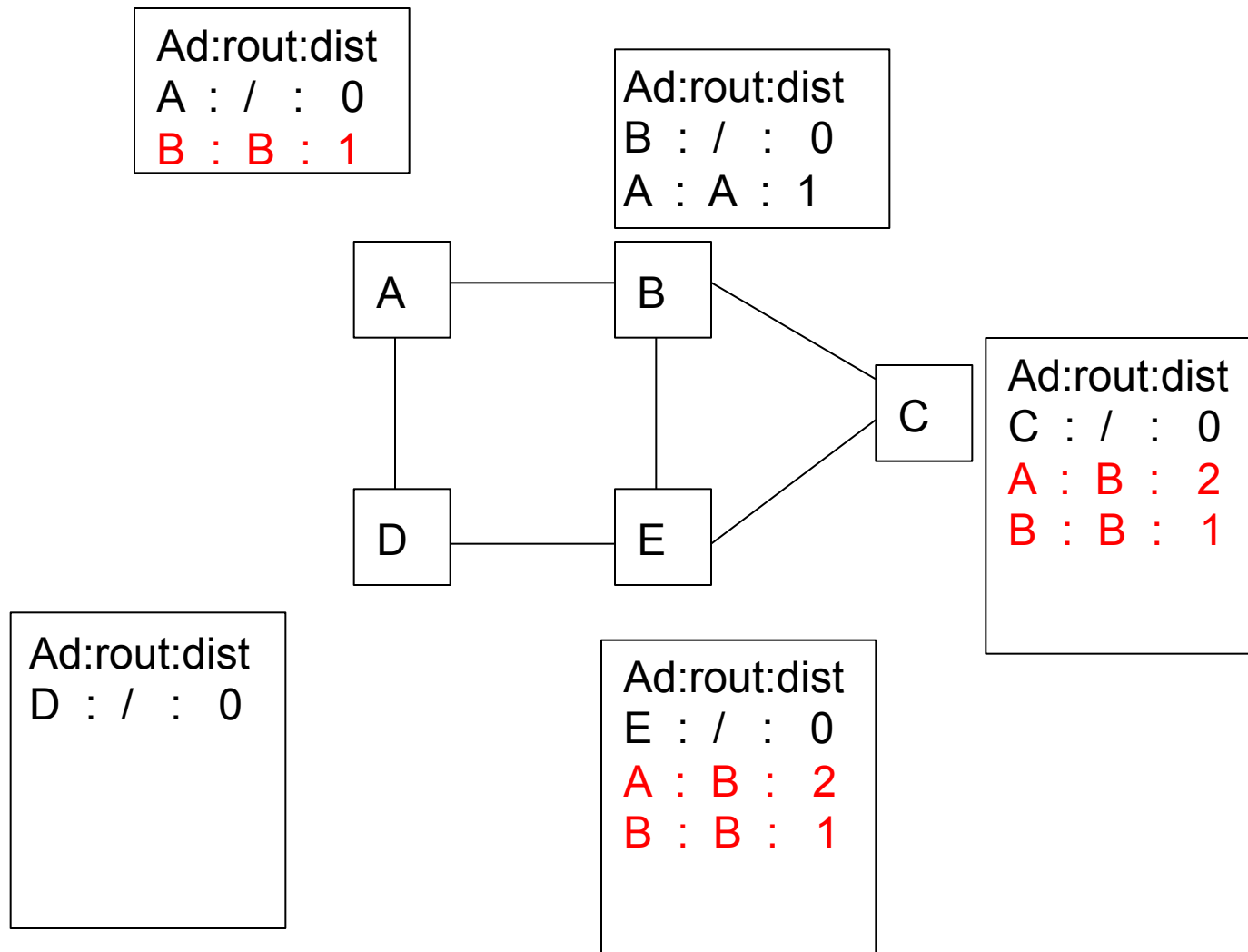


Ad:rout:dist  
C : / : 0

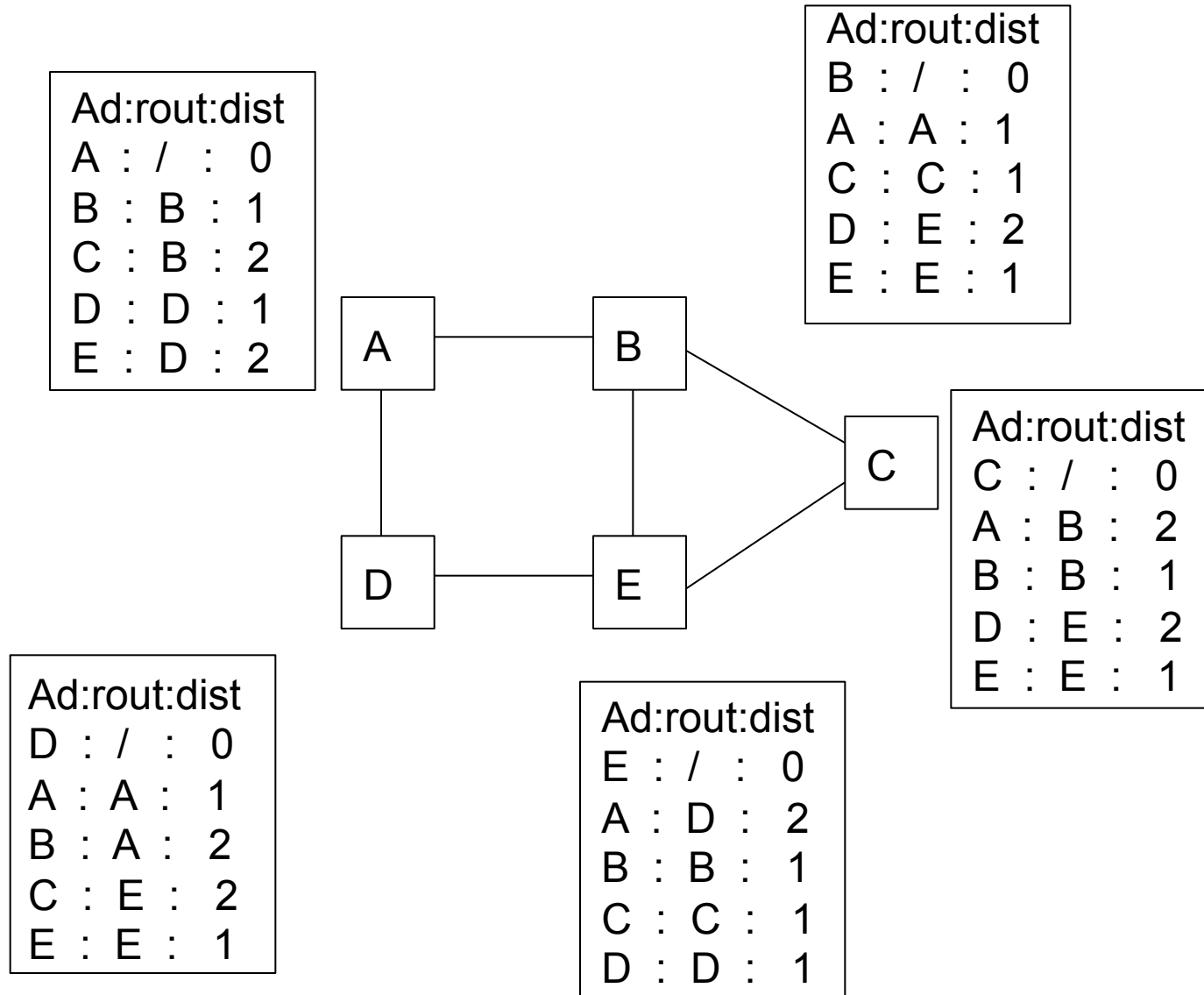
Ad:rout:dist  
D : / : 0  
A : A : 1

Ad:rout:dist  
E : / : 0

# Example



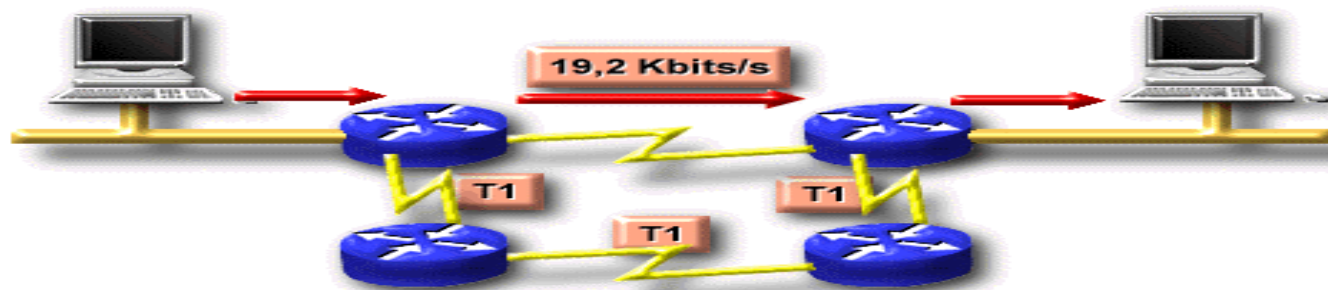
# Example



# Routing Information Protocol (RIP)

- ◆ RIP est un protocole de routage à vecteur de distance
- ◆ Métrique (nombre de sauts), maximum 15
- ◆ Mise à jour diffusée toutes les 30 secondes
- ◆ Deux versions:
  - ◆ RIP-1, rfc 1058, 1988.
  - ◆ RIP-2, rfc 1388, 1993
- ◆ RIPv1 atteint l'ensemble de ses voisins en envoyant des mises à jour à l'adresse IPv4 de tous les hôtes, 255.255.255.255, par le biais d'une diffusion.

## Aperçu du protocole RIP



● La métrique du nombre de sauts sélectionne le chemin.

T1: 1,5 Mb/s

# Routing Information Protocol (RIP)

## ◆ RIP permet:

- détection automatique de réseaux (meilleur chemin).
- mise à jour et maintenance des tables de routage (meilleur chemin).

## ◆ Meilleur chemin: La détermination du meilleur chemin d'un routeur implique d'évaluer plusieurs chemins menant au même réseau de destination et de choisir le chemin le plus court pour atteindre ce réseau.

## ◆ RIP sélectionne le meilleur chemin dont le coût le plus faible (le nombre de routeurs entre un routeur et le réseau de destination).

# Routing Information Protocol (RIP)

- ◆ L'indication entre crochets [**x/y**]: **x**: nombre représente la Distance Administrative ou fiabilité du protocole utilisé (120 pour RIP). **y**: chiffre est la métrique
- ◆ Un routeur RIP peut être configuré comme **actif** ou **passif**.
- ◆ Un routeur **actif** écoute les messages RIP, met à jour sa table de routage et **émet ses propres messages RIP**.
- ◆ Un routeur **passif** écoute les messages de routage, met à jour sa table de routage mais **n'émet aucun message RIP**.
- ◆ Un message RIP peut être une requête ou une réponse.

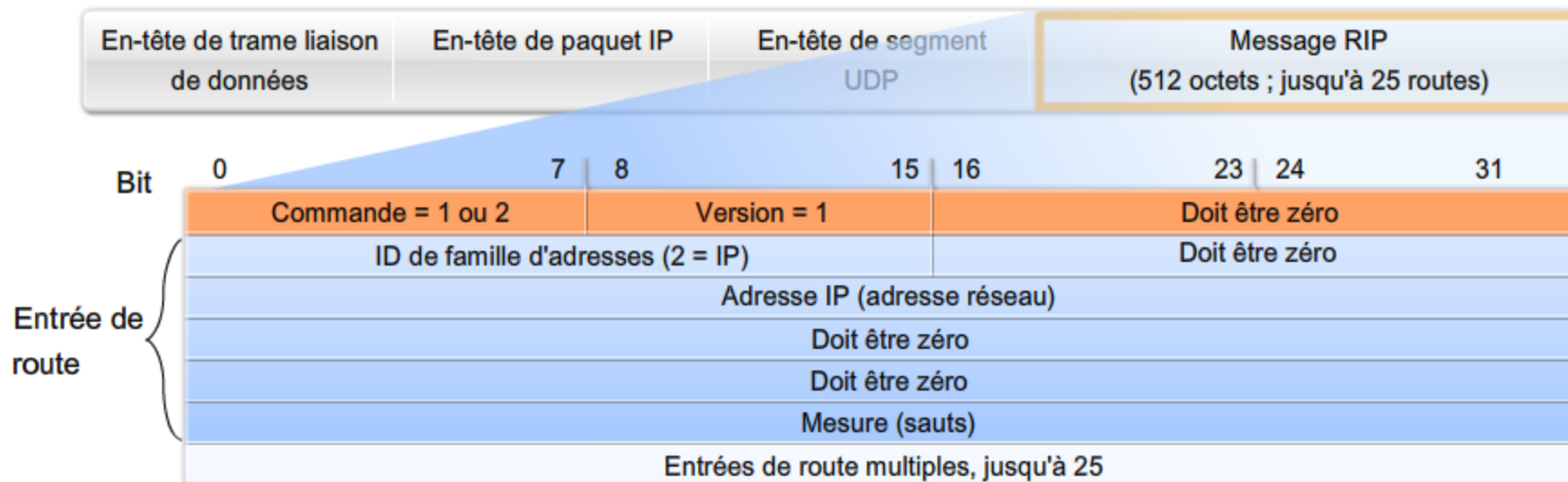
# Format du message RIPv1

Message RIPv1 encapsulé



- ◆ La partie données d'un message RIP est encapsulée dans un segment UDP, avec les numéros de ports source et de destination définis sur 520.
- ◆ L'en-tête IP et les en-têtes de liaison de données ajoutent des adresses de destination de diffusion avant l'envoi du message à toutes les interfaces configurées RIP.
- ◆ Chaque interface configurée sous RIP envoie un message de requête au démarrage, demandant à ce que tous les voisins RIP envoient leurs tables de routage complètes. Un message de réponse est renvoyé par les voisins RIP.

## Format de message RIPv1



Champ	Description
<b>Commande</b>	1 pour une demande ou 2 pour une réponse.
<b>Version</b>	1 pour RIP v 1 ou 2 pour RIP v 2.
<b>ID de famille d'adresses</b>	2 pour IP et 0 si la demande concerne la table de routage complète.
<b>Adresse IP</b>	Adresse de la route de destination, qui peut être un réseau, un sous-réseau ou une adresse d'hôte.
<b>Mesure</b>	Nombre de sauts compris entre 1 et 16. Le routeur émetteur augmente la mesure avant d'envoyer le message.



# RIP: Configuration

## ◆ Configuration de base:

### 1. On active le routage RIP

- Router(config)# router rip
- Router (config-router)# version 1 (version (1 ou 2)

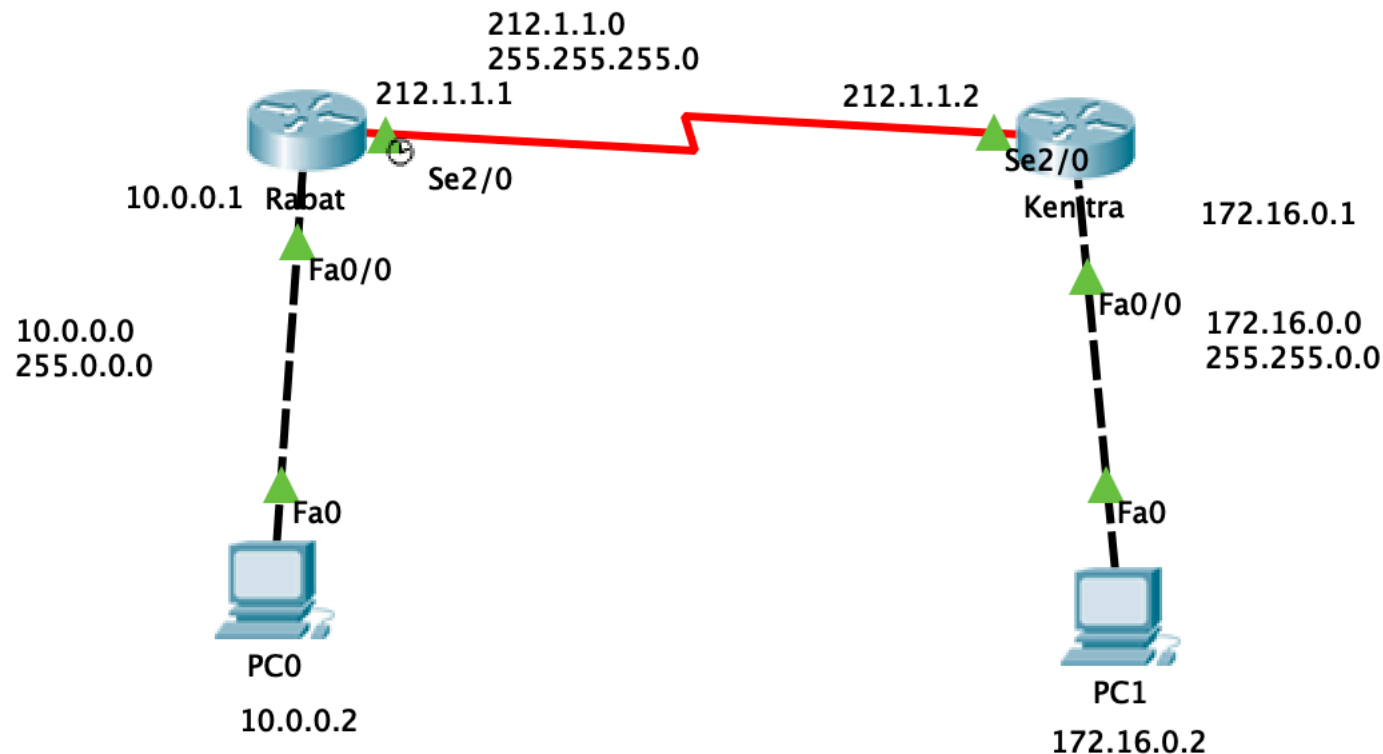
### 2. Annoncer les réseaux directement connectés (réseaux à transmettre dans les messages RIP aux routeurs voisins)

- Router(config-router)# network @IP Réseau
- Router(config)# ip default-network @IP Réseau (permet de faire propager une destination par défaut)

### 3. Vérification

- Router# show ip protocols: (vérifie que le routage RIP est configuré, que les interfaces envoient et reçoivent des mises à jour RIP et que le routeur annonce les réseaux appropriés.)

# Exemple (routage dynamique)



# Exemple

## Configuration des interfaces du routeur

- Rabat(config)#int fastEthernet 0/0
- Rabat(config-if)#ip address 10.0.0.1 255.0.0.0
- Rabat(config-if)#no sh
- Rabat(config)#int s2/0
- Rabat(config-if)#ip address 212.1.1.1 255.255.255.0
- Rabat(config-if)#no sh

## On active le routage RIP

- Rabat(config)# router rip
- Rabat(config-router)# version 1

## Annoncer les réseaux directement connectés

- Rabat(config-router)# network 10.0.0.0
- Rabat(config-router)# network 212.1.1.0

# Example

## 3. Vérification

- Rabat# `show ip protocols`
- Routing Protocol is "rip"
- Sending updates every 30 seconds, next due in 0 seconds
- Invalid after 180 seconds, hold down 180, flushed after 240
- Default version control: `send version 1`, receive any version
- Routing for Networks:
- `10.0.0.0`
- `212.1.1.0`
- Distance: (default is 120)

# Example

## 3. Vérification

➤ Rabat# `show ip route`

Codes: C - connected, S - static, I - IGRP, **R - RIP**,

- C 10.0.0.0/8 is directly connected, FastEthernet0/0
- **R\* 172.16.0.0/16 [120/1] via 212.1.1.2, 00:00:18, Serial2/0**
- C 212.1.1.0/24 is directly connected, Serial2/0

# Exemple

## Configuration des interfaces du routeur

- Kenitra(config)#int fastEthernet 0/0
- Kenitra(config-if)#ip address 172.16.0.1 255.255.0.0
- Kenitra(config-if)#no sh
- Kenitra(config)#int s2/0
- Kenitra(config-if)#ip address 212.1.1.2 255.255.255.0
- Kenitra(config-if)#no sh

## On active le routage RIP

- Kenitra(config)# router rip
- Kenitra(config-router)# version 1

## Annoncer les réseaux directement connectés

- Kenitra(config-router)# network 172.16.0.0
- Kenitra(config-router)# network 212.1.1.0

# Example

## 3. Vérification

- Kenitra# `show ip protocols`
- Routing Protocol is "rip"
- Sending updates every 30 seconds, next due in 0 seconds
- Invalid after 180 seconds, hold down 180, flushed after 240
- Default version control: `send version 1`, receive any version
- Routing for Networks:
- `172.16.0.0`
- `212.1.1.0`
- Distance: (default is 120)

# Example

## 3. Vérification

➤ Kenitra# `show ip route`

Codes: C - connected, S - static, I - IGRP, **R - RIP**,

- **R 10.0.0.0/8 [120/1] via 212.1.1.1, 00:00:17, Serial2/0**
- C 172.16.0.0/16 is directly connected, FastEthernet0/0
- C 212.1.1.0/24 is directly connected, Serial2/0



# Route par défaut

## 3. Ajouter uniquement la route par défaut dans un routeur

Puis la rendre dynamique via la commande **default-information originates**

- Kenitra(config)#ip route 0.0.0.0 0.0.0.0 212.217.10.1
- Puis la rendre dynamique via la commande **default-information originates**
- Kenitra(config)#router rip
- Kenitra(config-router)#default-information originates



# RIPv1

- Réseaux discontinues séparés par des routeurs
- RIPv1 résumé automatiquement à la frontière des réseaux discontinues **les réseaux qu'ont même masque**.
- **Empêcher d'envoi des paquets sur une interface**, on utilise la commande **#passive-interface Fa0/0**
- Le routeur peut regrouper plusieurs routes dans une route avec la commande **#automatic network summurization (max=4)**
- Dans un réseau avec plusieurs routeurs, pour se connecter à l'internet, il faut ajouter une route par défaut dans toutes les routeurs (statique)
- Sinon, on ajoute la route par défaut uniquement dans le routeur de frontière, et on la rendre dynamique via la commande **#default-information originates** (dynamique).
- Envoie de toutes les routes statiques: **#redistribute-static**

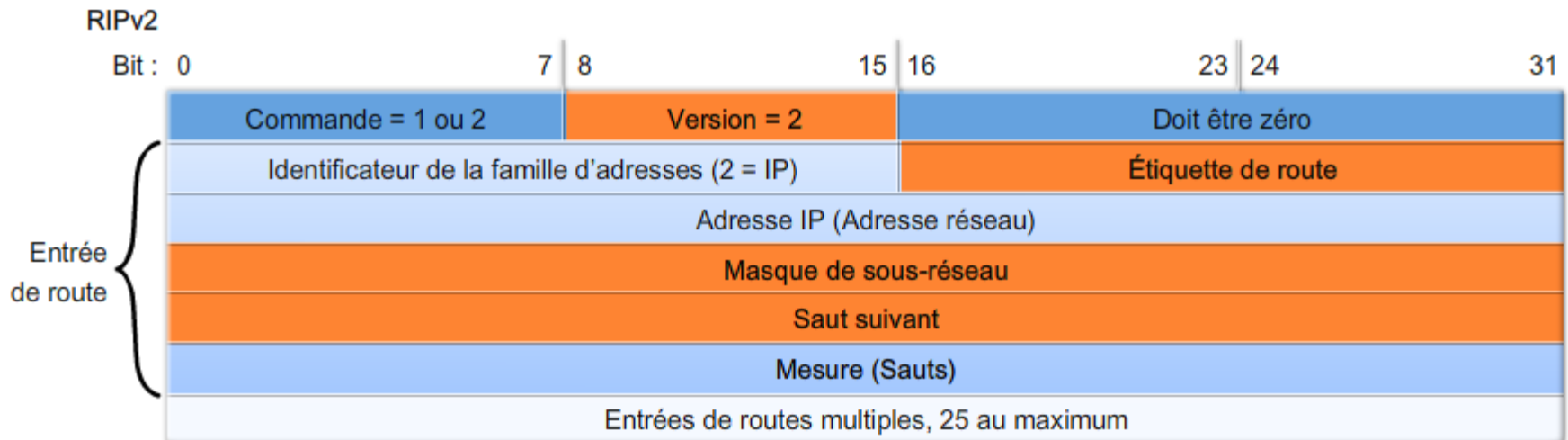
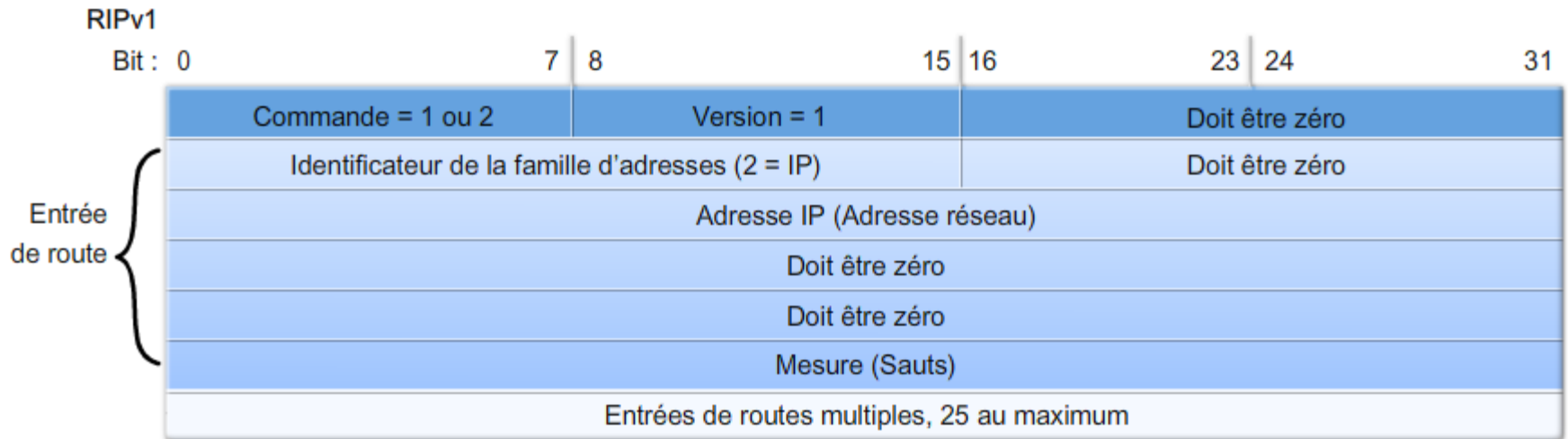
# RIPv1 (suppression de routes)

- Le routeur RIP actif permet aux autres routeurs de mettre à jour leurs tables de routage toutes les 30 secondes.
- Si un routeur ne reçoit aucune mise à jour d'un autre routeur dans un délai de 180 secondes, il marque les routes desservies par ce dernier comme inutilisables.
- S'il n'y a aucune mise à jour après 240 secondes, le protocole RIP supprime toutes les entrées correspondant au routeur qui ne répond pas.
- Chaque message de diffusion RIP contient des paires adresses IP/ nombre de routeurs à traverser (ou nombre de sauts).

# RIPv2

- RIPv1 ne peut pas prendre en charge la méthode VLSM.
- RIPv1 ne peut pas prendre en charge les routes CIDR qui sont des routes résumées avec un masque de sous-réseau plus petit que le masque par classe de la route. RIPv1 ignore ces super-réseaux dans la table de routage et ne les inclut pas aux mises à jour vers d'autres routeurs.
- RIPv2 étant un protocole de routage sans classe. RIPv2 inclut maintenant tous les sous-réseaux et leurs masques appropriés dans ses mises à jour de routage (no auto-summary).
- RIPv2 prend en charge VLSM pour propager tous les réseaux avec leurs masques de sous-réseau appropriés.
- RIPv2 prend également en charge les super-réseaux CIDR.
- RIPv2 est encapsulé dans un segment UDP via le port 520 et peut transporter jusqu'à 25 routes.

## Comparaison des formats des messages RIPv1 et RIPv2



# Echange des mise à jour

## ◆ Envoi de mises à jour de routage:

- Les mises à jour pendant toutes les 30 secondes sur les interfaces du routeur.
- Envoie des tables de routage entières dans les MAJ (**dans un paquet IP, une source et une destination**).
- **RIPv1** envoie à l'adresse de diffusion (**Broadcast** **255.255.255.255** (**les hôtes du réseau LAN lisent le paquet**) **via les interfaces actifs du routeur**).
- **RIPv2** envoie à l'adresse de **Multicast 224.0.0.9** (groupe de routeurs RIPv2). Ce qui permet de réduire la charge de traitement des messages de MAJ sur les hôtes du même réseau.
- MAJ permet au routeur d'être à jour (table de routage à jour).

# Boucle de routage

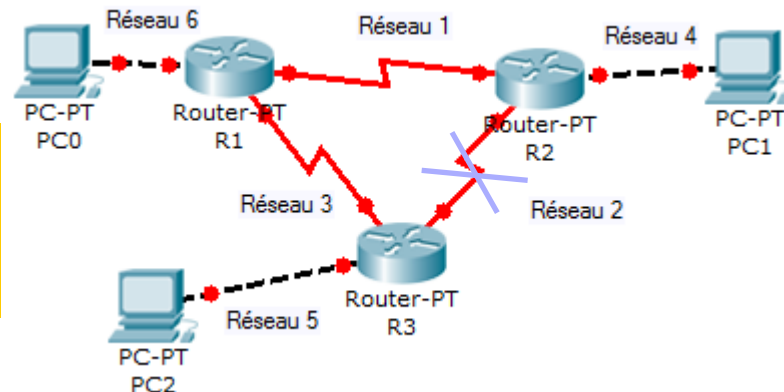
## ◆ Boucle de routage infinie:

- **Condition:** Un paquet IP transmis en continu entre une série de routeurs sans qu'il arrive à la destination.
- **Cas de boucle:** des routeurs échangent des MAJ avec des informations fausses (routes) (le réseau destinataire est inaccessible).
  - Une route statique mal configurée
  - Suppression d'un lien vers un autre réseau
  - Table de routage n'a pas mis à jour
- Le routeur peut supprimer le paquet IP à un certain moment si son TTL = 0 (durée de vie dont la valeur est diminuée de 1 à chaque routeur).

# Adaptation dynamique de RIP

- **Changement dynamique des tables de routage** des routeurs suite aux évènements:
  - a. **Ajout** d'un réseau ou d'un routeur;
  - b. **Suppression** d'un réseau ou d'un routeur;
  - c. **Coupure** d'un lien d'un routeur.
- **Suppression de ligne** grâce au timer associé à chaque ligne de la table de routage;
- **Changement des chemins** suivant les nouveaux coûts;
- **Exemple:**

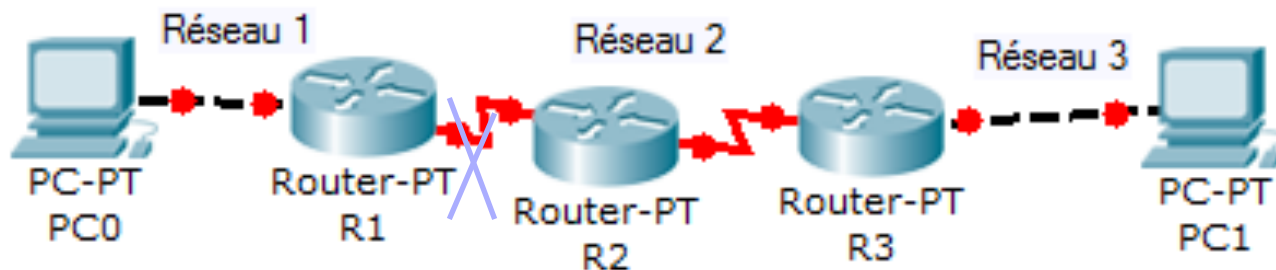
Donner les tables de routages dans le cas où le lien entre R3 et R2 est rompu.





# Problème du comptage à l'infini

- Que se passe-t-il sur le réseau suivant dans le cas où la ligne de R1 vers R2 est coupée ?



- **Solution:** Les paquets RIP ne contiennent pas toute la table de routage. Ils ne contiennent que les adresses qui n'ont pas été apprises par la ligne sur laquelle ils sont émis. **R1 envoie une route à R2 et dit à lui de ne pas me retourner cette information.**

# Boucle de routage

## ◆ Eviter la boucle de routage infinie:

- Charger les routeurs du traitement des messages inutiles (consommation de la bande passante du lien entre les routeurs).
- Le processus du routeur est chargé de traité des paquets inutiles.

## ◆ Solution de la boucle de routage infinie:

- Horizon partagé (split horizon)
- Métrique infinie
- Route empoisonnée
- Mise à jour déclenché

# Solution du problème du comptage à l'infini

## ➤ Split-Horizon (horizon partagé)

Une règle qui interdit à tout routeur d'annoncer un préfixe réseau via l'interface par laquelle il a appris l'existence de celui-ci ou par laquelle il passe pour accéder à ce réseau. **Un routeur ne propage pas les couples (routeur, distance) provenant d'un lien sur ce lien.** Ce mécanisme a été introduit dans RIP version 2.

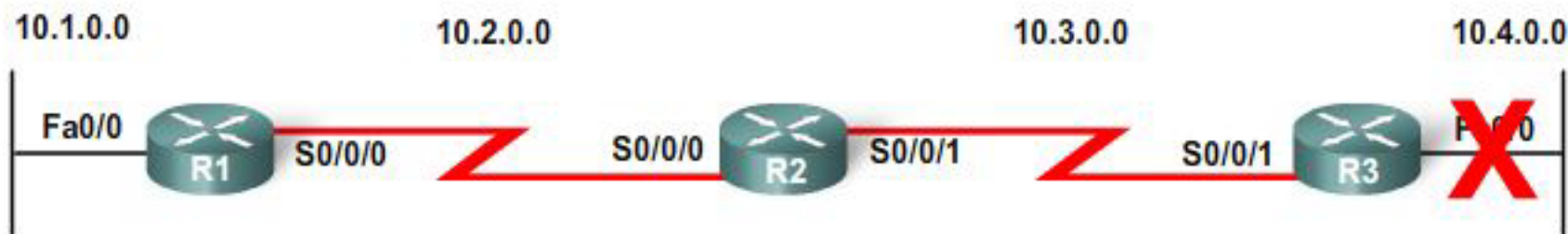
## ➤ Poison-Reverse (route empoisonnée)

Le routeur ne propage pas les couples (routeur, distance) provenant d'un lien sur ce lien en donnant une valeur **infinie aux distances**. Le routeur **ne supprime pas** la ligne immédiatement mais lui associe un coût infini (16) (inaccessible) qui sera propagé dans les prochains paquets RIP.

## ➤ Mise à jour déclenchée

Un vecteur de distance est envoyé aux routeurs voisins toutes les  $T1$  secondes. Un lien est considéré comme rompu par un routeur lorsque celui-ci n'a reçu aucun vecteur de distance depuis  $T2 = nT1$  secondes. **Ce routeur envoie alors immédiatement son vecteur de distance modifié sans attendre l'expiration de  $T1$ .** En cas de modification de la table suite à la réception d'un paquet RIP, la modification est envoyée immédiatement.

# Boucle de routage (exemple)



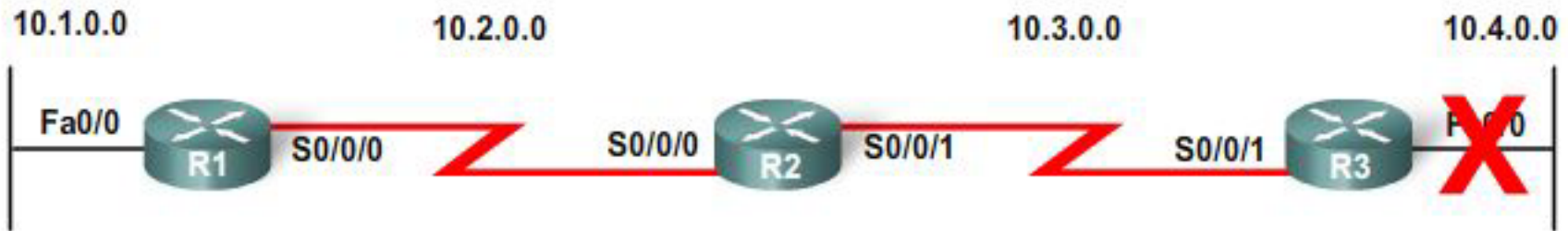
Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

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

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

R3 détecte que la liaison vers le réseau 10.4.0.0 est rompue.  
Une MAJ est envoyée par le routeur R2 contenant une route vers ce réseau.  
R2 ajoute cette route dans sa table.

# Boucle de routage (suite)



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
10.1.0.0	S0/0/0	1
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/1	1

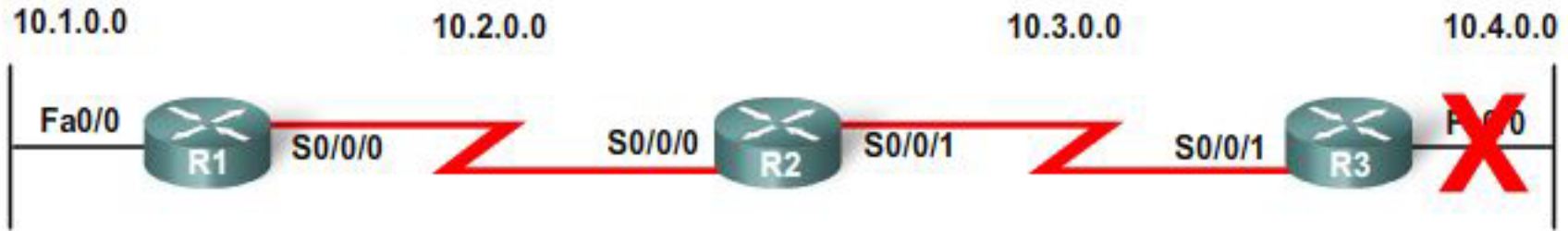
Réseau	Interface	Saut
10.1.0.0	S0/0/1	2
10.2.0.0	S0/0/1	1
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/0	1+1 = 2

R3 détecte que la liaison vers le réseau 10.4.0.0 est rompue.

Une MAJ est envoyée par le routeur R2 contenant une route vers ce réseau.

R2 ajoute cette route dans sa table.

# Boucle de routage



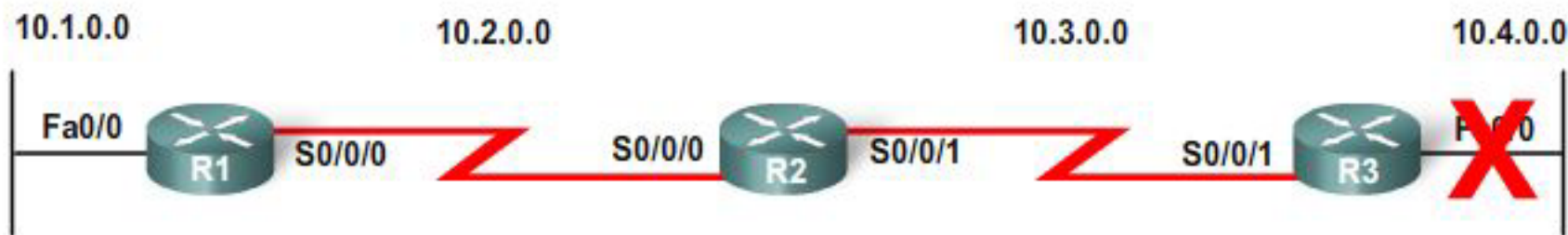
Réseau	Interface	Saut
10.1.0.0	Fa0/0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
10.4.0.0	S0/0/0	2

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

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

R1 envoie sa table de routage à R2. R2 envoie sa table à R3. Puis R3 envoie R2. Ainsi de suite. Ce qui donne la naissance de boucle. Le paquet IP va être perdu lorsque TTL devient 0.

# Boucle de routage (compteur infinie)



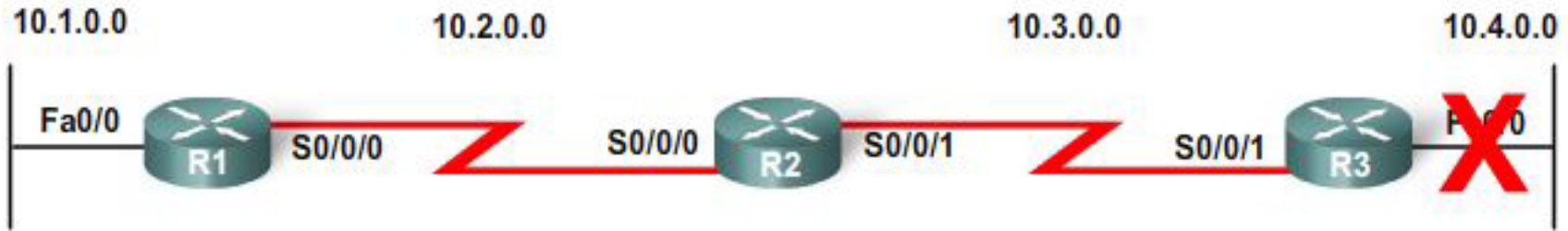
Réseau	Interface	Saut
10.1.0.0	Fa0/0	0
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/0	1
<del>10.4.0.0</del>	<del>S0/0/0</del>	<del>16</del>

Réseau	Interface	Saut
10.1.0.0	S0/0/0	1
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
<del>10.4.0.0</del>	<del>S0/0/1</del>	<del>16</del>

Réseau	Interface	Saut
10.1.0.0	S0/0/1	2
10.2.0.0	S0/0/1	1
10.3.0.0	S0/0/1	0
<del>10.4.0.0</del>	<del>S0/0/0</del>	<del>16</del> (infinie)

R1 envoie sa table de routage à R2. R3 envoie sa table à R2.  
R2 envoie sa MAJ à R1. Puis à R3. Lorsque la valeur infinie est arrivée, le routeur détruit le paquet (la route est inaccessible).

# Horizon partagé



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
10.1.0.0	S0/0/0	1
10.2.0.0	S0/0/0	0
10.3.0.0	S0/0/1	0
10.4.0.0	S0/0/1	1

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

## Règle:

R1 n'envoie pas à R2 les réseaux envoyés par R2. R1 envoie uniquement 10.1.0.0.

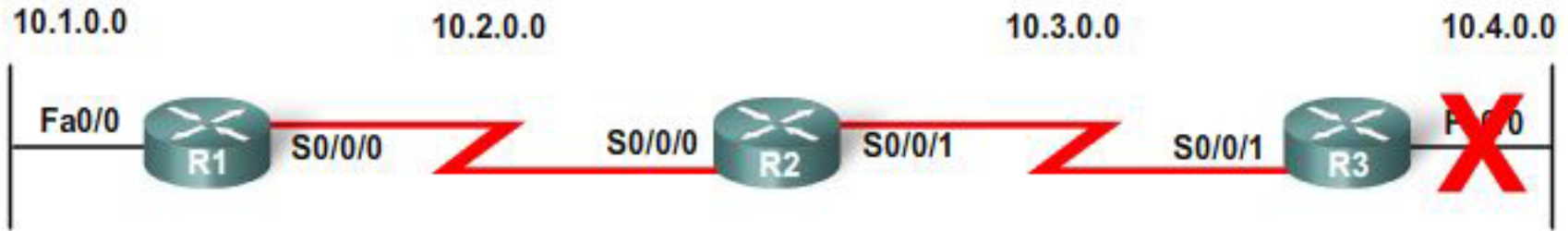
R2 envoie à R1 uniquement 10.3.0.0 et 10.4.0.0.

R2 envoie à R3 uniquement 10.1.0.0, 10.2.0.0.

R3 envoie à R2 uniquement 10.4.0.0.



# Route empoisonnée



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	16

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

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

R3 détecte que le réseau 10.4.0.0 est inaccessible. Il change la métrique à sa valeur infinie (16). Ensuite, il envoie sa table de mise à jour. Les routeurs R2, R1, marquent infinie pour le même réseau.