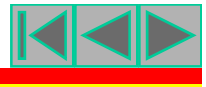


Cours RSX103 - Chapitre 05

Routage IP et Dans Internet

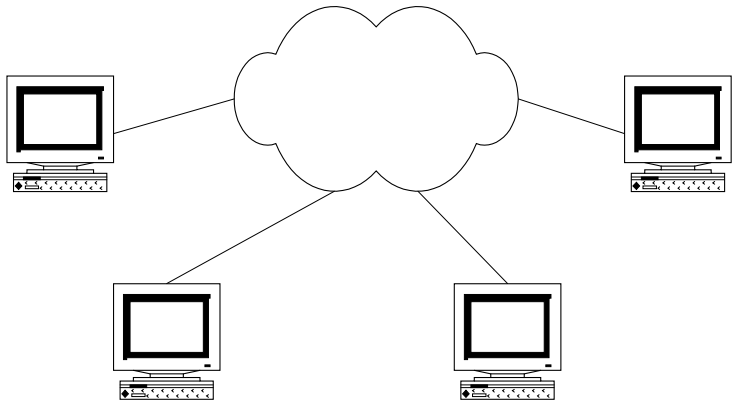
« première partie »

Utilité de la couche liaison de données

- **La couche liaison de données nous permet de résoudre deux problèmes :**

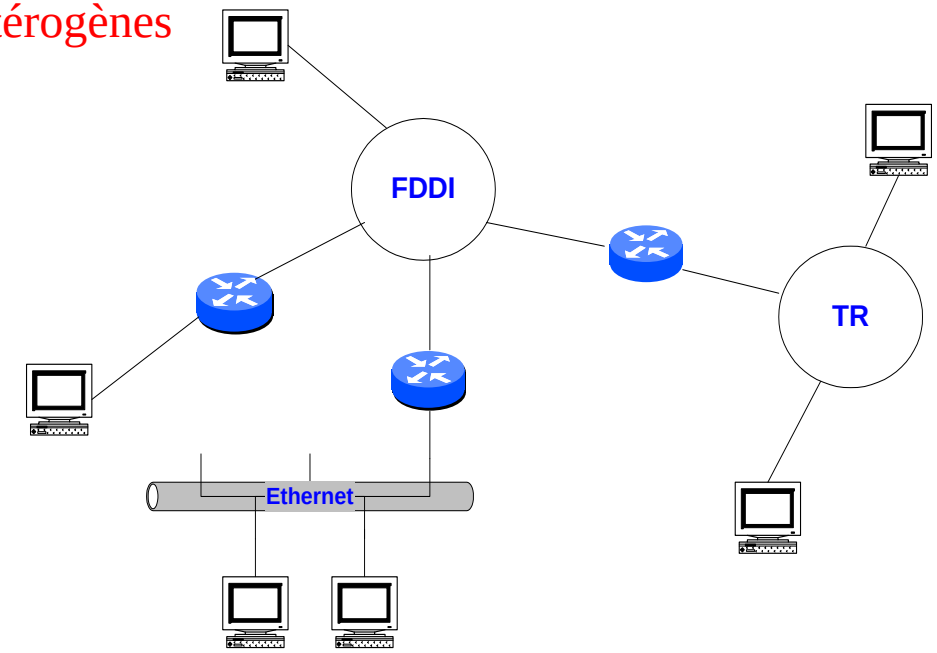
1. **Transfert fiable de trames entre deux machines connectées par "un même fil"**
   - Utile surtout pour les réseaux longue distance ou les réseaux téléphonique (protocoles HDLC, PPP) **avec des acquittement** et **numérotation des trames**, …etc.

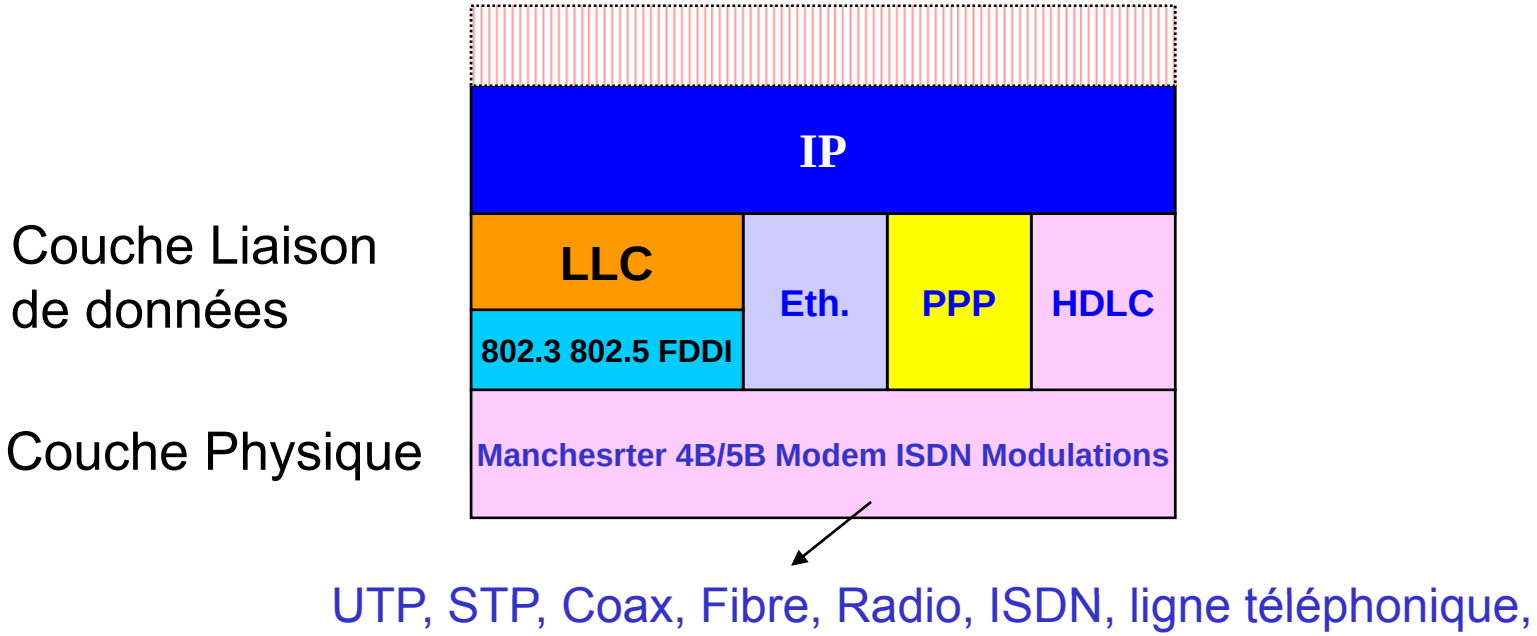
2. **Construire des réseaux locaux**
   - Ethernet, Token Ring, FDDI
   - Permet un transfert de trames entre N machines connectées à un même milieu de transmission
     - Distance limitée



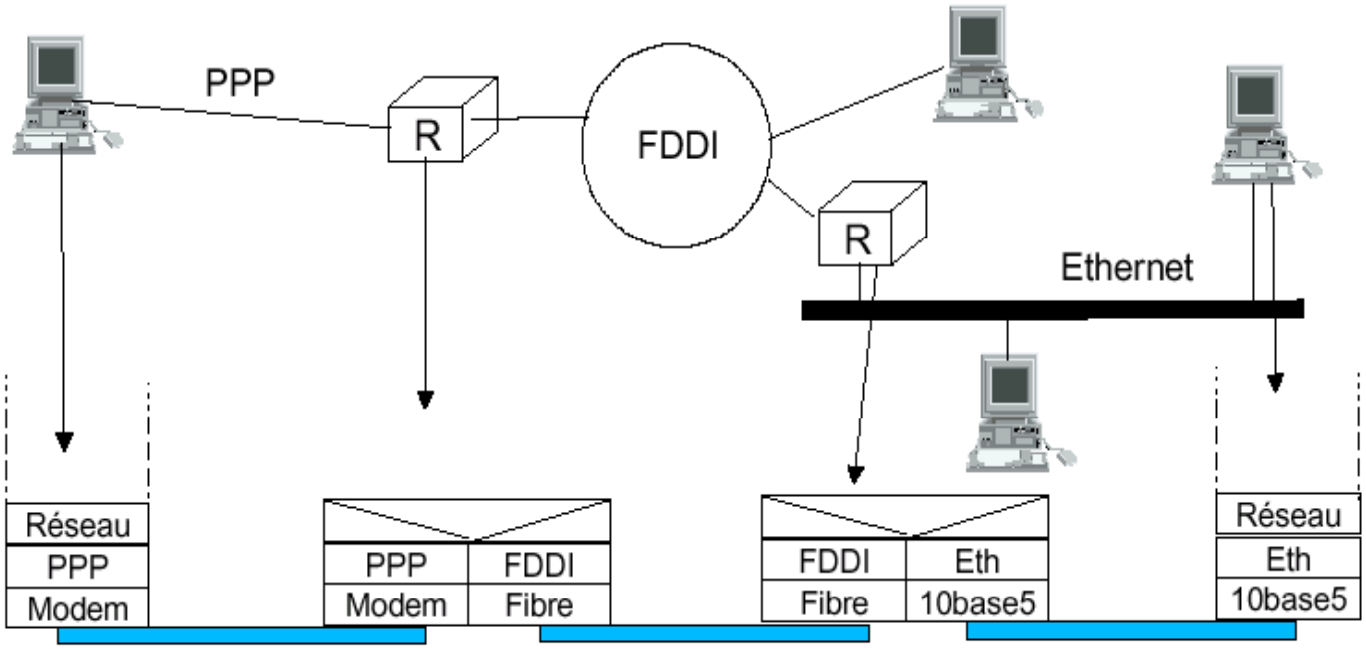
Limitations de la couche liaison de données

- **Couche liaison de données classique**
  - En général, connexion uniquement entre deux machines à travers un "même fil
- **Couche liaison de données, réseaux locaux**
  - Limites à l'étendue géographique du réseau
  - Limites sur le nombre de machines que l'on peut connecter au réseau local
- **Environnement hétérogènes ?**
  - En pratique, on va devoir transmettre des paquets sur une plus grande distance et à travers des réseaux beaucoup plus hétérogènes

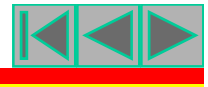




- Solution : Ajout d'une couche réseau
  - Objectifs de la couche réseau
    1. service de transmission de données de bout en bout à travers un ensemble de réseaux hétérogène
    2. service indépendant des détails de la couche 2 (exemple : bits A et C dans token ring)



- Pour réaliser ses services : la couche réseau va s'appuyer sur des **relais intermédiaire** qui vont fonctionner **à l'intérieur de la couche réseau** « Routeurs »
- **Définition :**
  - Un routeur est un relais au niveau de la couche réseau : un élément qui **dispose de plusieurs interfaces** avec **plusieurs connexions à des réseaux différents** (réseaux Etherent différents ou sur des réseaux hétérogènes)



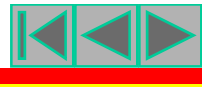
- Fonctionnement de la couche réseau : Principes de base

1. chaque station/routeur connecté au réseau doit être identifié par une "adresse réseau" qui va être construite de manière toute a fait indépendante du type de couche liaison de données sous jacente

   ⇒ On va avoir un espace d'adresses indépendant des adresses de la couche liaison de données qui est construit sur 48 bits »

2. la couche réseau achemine les paquets depuis une source vers une destination via de nombreux routeurs intermédiaires :

   - le service fourni doit être complètement indépendant du type de couche liaison de données sous jacente
   - l'utilisateur de la couche réseau ne doit rien connaître de l'organisation interne de la couche réseau pour pouvoir l'utiliser



♦ Quel type de service pour la couche réseau ?

Historiquement deux services sont fournies au niveau de la couche réseau (philosophie Internet)

1 - sans connexion et non fiable

- Solution choisie par l'Internet
- motivation : limiter la complexité de la couche réseau
  - ✓ se concentrer sur l'acheminement des paquets de la source vers une destination)
  - ✓ Les routeurs intermédiaires ne vont pas s'occuper de la qualité de transmission. On laissera cette tâche à la charge des S & D.

❖ un service sans connexion non fiable implique que :

1. des données peuvent se perdre
2. des données peuvent être réordonnées (paquets A,B arrivent dans l'ordre B,A)
3. des données peuvent être dupliquées (A arrive A, A)

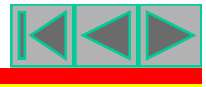
- la récupération des erreurs devra se faire par une couche au-dessus de la couche réseau (couche transport)

2. Service orienté connexion fiable (rendre le meilleur service possible aux utilisateurs

- ❖ Vision des opérateurs de téléphonie quand ils ont commencé à faire des réseaux de transmission de données (X25,…)
- (idée de base : conserver le même mode de fonctionnement que dans les réseaux téléphonique
- ❖ Propriétés désirées des connexions

1. phase préalable d'ouverture de connexion (on peut faire payer à la durée de communication)
   - ⇨ on peut associer à une connexion des paramètres importants tels que coût et qualité d'une connexion
2. communication bidirectionnelle et respect de la séquence
3. contrôle de flux possible pour éviter de surcharger le réseau ou le receveur

❑ Dans un service orienté connexion, on ajoute de la complexité dans la couche réseau pour simplifier les couches supérieures (l'inverse de l'internet)



- **Comment construire une couche réseau ?**

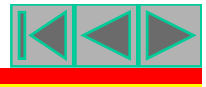
  - deux **organisations internes** possibles

1. datagrammes :

- les paquets vont contenir une **AS** et une **AD**
- tous les routeurs vont acheminer les paquets en consultant **l'adresse de destination** contenue dans le paquet (modèle de la poste)

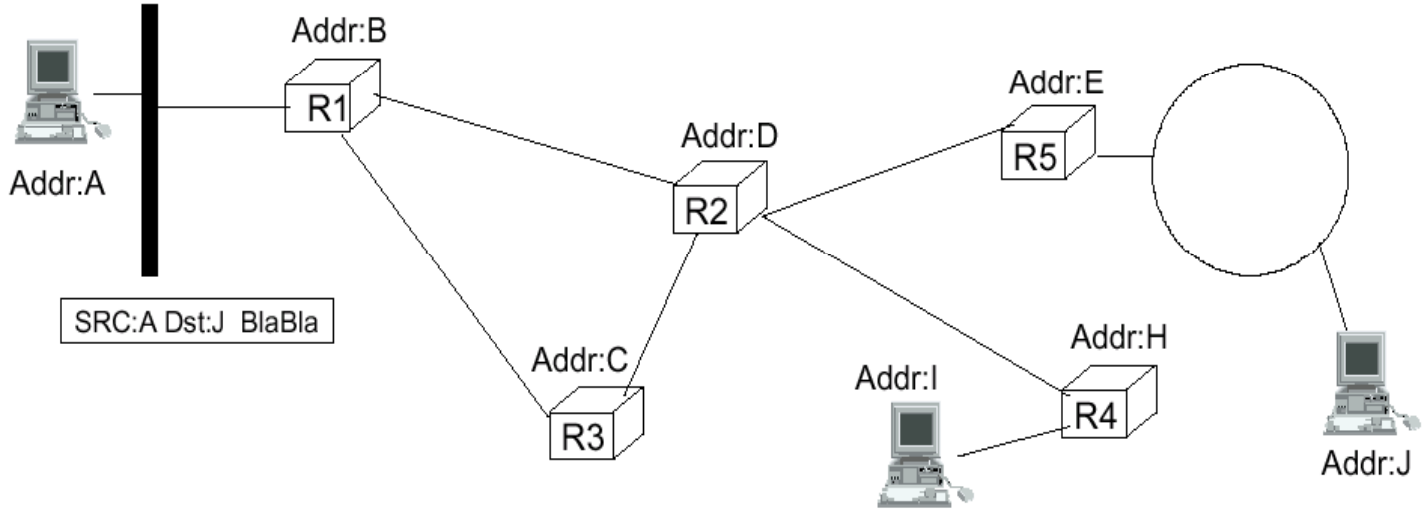
2. circuits virtuels :

- on va établir une **association logique** à l'intérieur du réseau pour acheminer les informations, comme on fait dans les réseaux téléphoniques

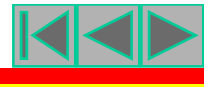


- Comment construire une couche réseau ?
  - En théorie, l'organisation interne de la couche réseau est indépendante du service fourni par la couche réseau,
  - On pourrait offrir un service orienté connexion fiable sur base d'une organisation en datagramme ou sur base d'une organisation en circuits virtuels ou l'inverse.
  - En pratique :
    - ✓ Le service sans connexion non fiable s'appuie sur une organisation en datagramme : c'est l'Internet.
    - ✓ Le service orienté connexion fiable (le plus populaire X25 s'appuie) sur un réseau en circuit virtuel.

Réseau à datagrammes ?



1. Chaque station du réseau est identifiée par son adresse (unicité des adresses)
2. Chaque paquet contient :
   - ✓ adresse réseau (niveau réseau) de la source
   - ✓ adresse réseau (niveau réseau) de la destination
   - ✓ information utile
3. Chaque routeur analyse l'adresse de destination de tout paquet reçu et décide sur quelle ligne de sortie envoyer le paquet pour joindre destination



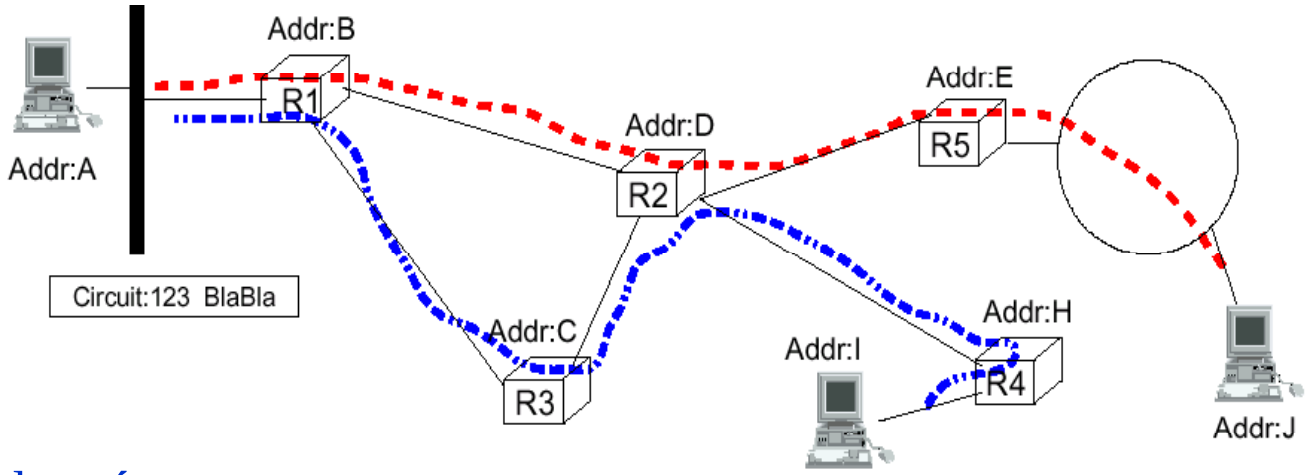
Objectif : Simplifier le fonctionnement des routeurs

⌦ le choix de la route sur base de l'adresse destination dans chaque paquet est complexe

Principe :

1. Etablir un circuit virtuel à travers le réseau avant de transférer les données ;
   - Cet établissement impose le choix d'une route à travers le réseau, mais le routage se fait par circuit et non pas par paquet comme avec un réseau à datagrammes
2. Une fois le circuit établi, *on peut transférer de l'information utile et tous les paquets d'un circuit suivront le même chemine dans le réseau*

Réseau à circuits virtuels (2)



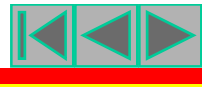
Contenu des paquets de données

1. identification du circuit virtuel
2. information utile

Comment identifier les circuits dans les paquets ?

- Première solution : (naïve) numéro unique pour chaque CV pour l'ensemble du réseau : pose deux problèmes
  1. difficile à coordonner le choix des numéros (car il faut trouver un mécanisme pour coordonner l'attribution des numéro des CV : *c'est facile s'il y un ordinateur central qui décide de l'attribution , dans le cas distribué c'est beaucoup plus compliqué*
  2. nombre total de circuits limité (selon le nombre de bits utilisés pour ça)

Réseau à circuits virtuels (3)



➢ Comment identifier les circuits dans les paquets ?

- Sur une ligne donnée, il suffit de pouvoir distinguer à quel circuit appartient chaque paquet passant sur cette ligne
- un numéro unique doit donc correspondre à chaque circuit sur chaque ligne du réseau
- mais un circuit virtuel ne doit pas avoir le même numéro sur tous les chemins qui composent un circuit donné du réseau

➢ Solution : Comment modifier le numéro unique de ligne en ligne ?

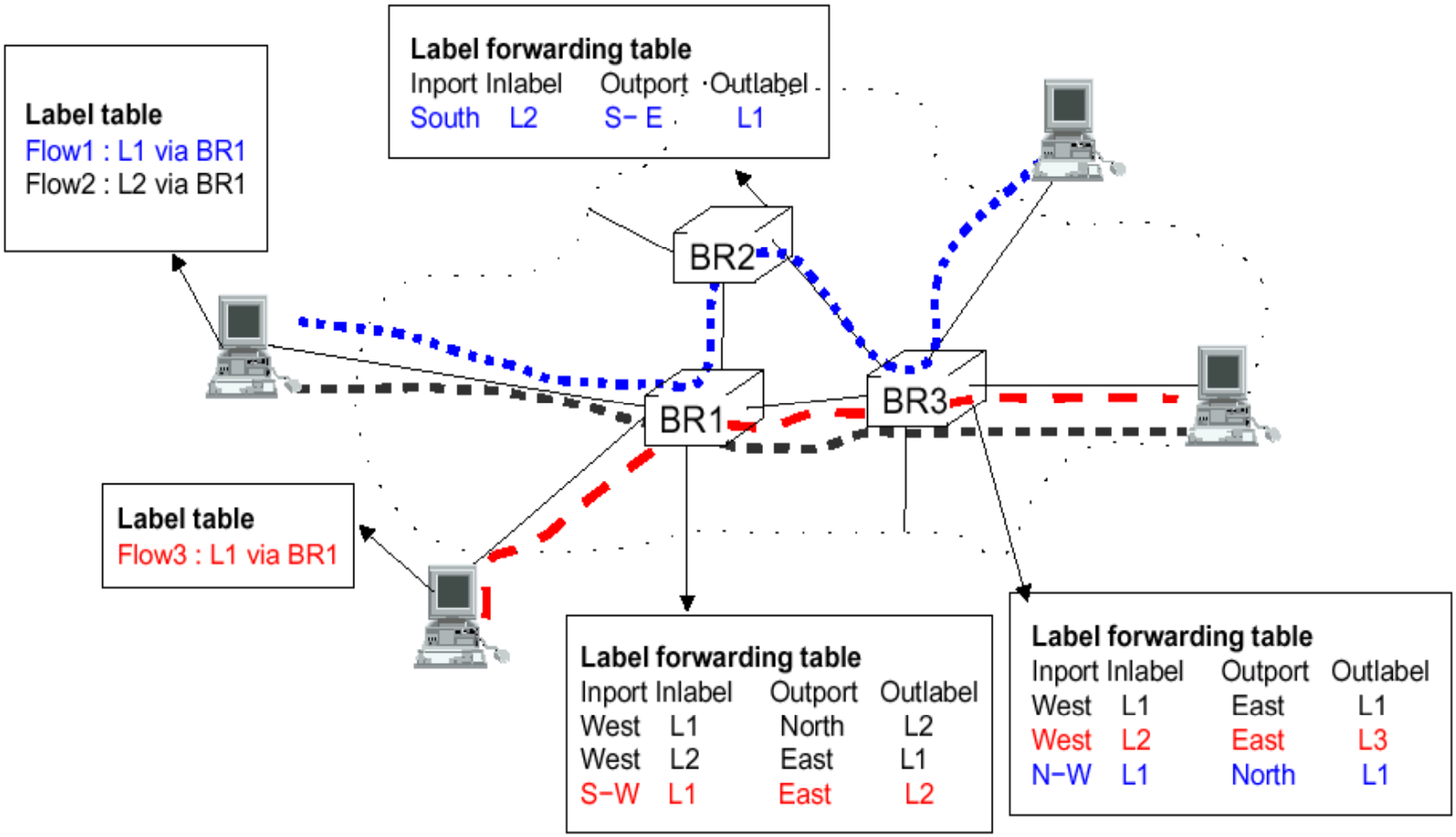
- Modification à réaliser dans les routeurs sur base d'une table

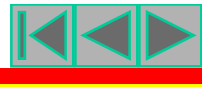
Label Forwarding Table

Port entrée	Numéro	Port Sortie	Numéro
South	L2	S- E	L1

•table est modifiée lors de chaque établissement ou fermeture de circuit virtuel

Réseau à circuits virtuels : exemple





1. Objectif principal de la couche réseau : « acheminer les paquets de la source à la destination »

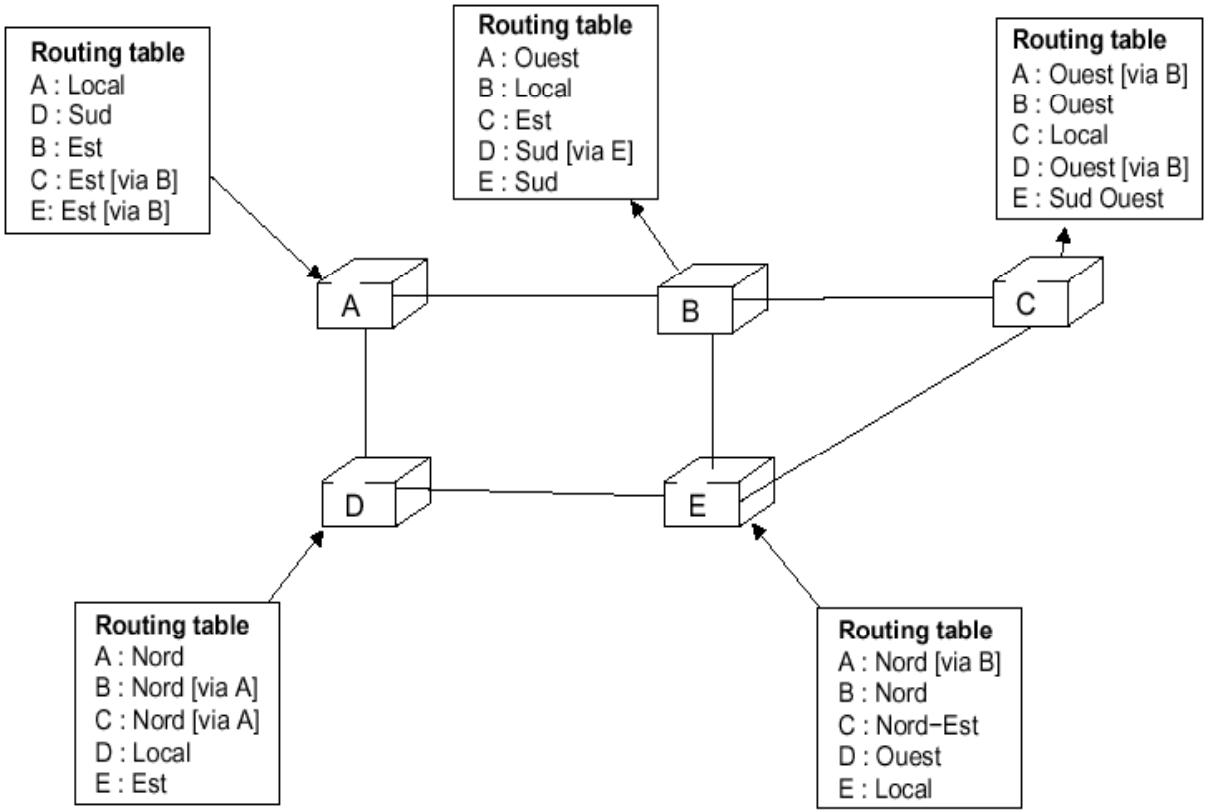
****** Routage *****

2. Dans un réseau, comment trouver le chemin d'un point A à un point B ?
   - Algorithme de routage utilisé dans les éléments intermédiaires (routeurs) du réseau
     1. Pour chaque paquet, ou demande d'ouverture de circuit virtuel reçu, *l'algorithme de routage décide via quel port de sortie le paquet ou la demande d'ouverture sera transmis*
     2. Fonctionne en général sur base d'une table de routage

Pour que les routeurs puissent remplir correctement ce service, il va falloir remplir correctement les tables de routage

Routage : exemple

- Ces tables de routage vont spécifier pour l'ensemble du réseau quel port « ou quelle interface » qu'il faut utiliser pour arriver à la destination

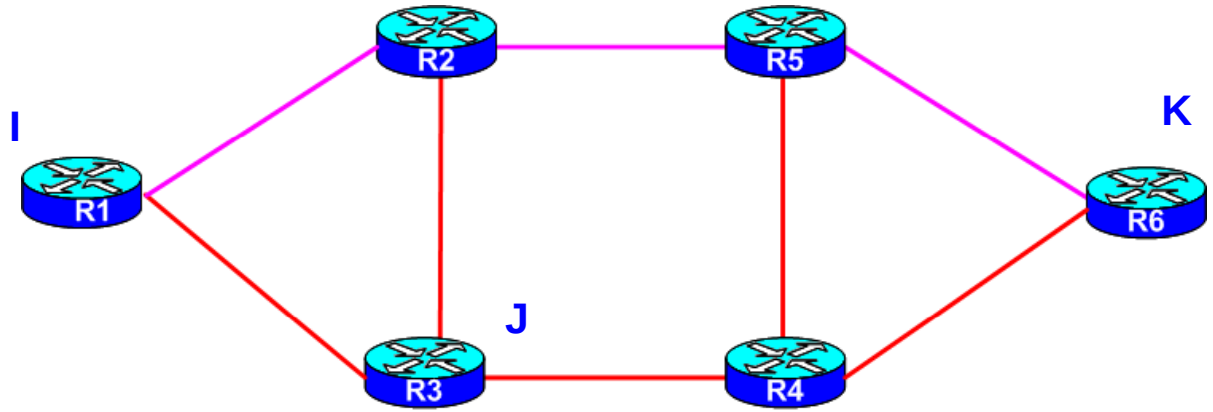


- Exemple : Dans le cas de routeur B :
  - Si AD = B alors le message est arrivé
  - Si AD = A alors il faut l'envoyer vers l'Ouest
  - Si AD = C alors il faut l'envoyer vers l'Est
  - Si AD = E alors il faut l'envoyer vers le Sud

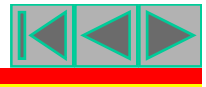
Si (AD = D) alors il faut l'envoyer vers le sud « via E »

- Principe d'optimalité :
  - Si le routeur J se trouve sur le meilleur chemin qui va du routeur I vers le routeur K, alors :

    le meilleur chemin de J à K suit la même route que le meilleur chemin de I à K
  - on peut donc trouver le meilleur chemin de façon incrémentale
  - Dans un réseau
    - l'ensemble des meilleurs chemins partants d'un routeur est un arbre dont la racine est ce routeur



Ça veut dire qu'on va se trouver dans un réseau ou la route qui est suivi par le message dépend uniquement de l'adresse de destination



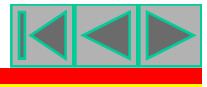
- Premier Problème : Meilleur chemin dans un réseau ? Comment le définir

1. plus court chemin [le moins de lignes à traverser] , en terme de nombre de routeurs intermédiaires à traverser
2. chemin avec le délai le plus court (peut contenir plus de routeurs que d'autres)
3. chemin avec le débit le plus élevé

- En général, on caractérise chaque ligne par un nombre [métrique]
  - on va essayer de chercher le chemin qui minimise la somme des nombres associées aux différentes liaisons qui composent le chemin.

- Deuxième problème : Comment construire les tables de routage ?

1. **Routage statique**
2. **Routage dynamique**



Principe

un ordinateur spécialisé calcule les tables des routage pour tous les routeurs du réseau

calcule des tables de routage

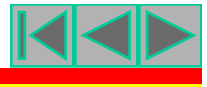
- algorithme pour trouver le chemin le plus court
- algorithmes plus complexes pour optimiser l'utilisation du réseau en fonction du trafic

Avantages du routage statique

1. facile à utiliser dans un petit réseau
2. optimisation possible des tables de routages (pour pouvoir par exemple, répartir l'ensemble du trafic à l'intérieur du réseau

Désavantages

1. pas d'adaptation dynamique à l'évolution du trafic
2. comment faire quand en cas de en panne ?



➢ Principe :

- les routeurs coopèrent pour mettre à jour leurs tables de routage de façon dynamique en s'appuyant sur un algorithme distribué
- utilisé dans quasiment tous les réseaux

Avantage

- Adaptabilité aux pannes de lignes
  - (en cas de panne, on pourra facilement informer l'ensemble des routeurs de façon à ce qu'ils mettent à jours leurs tables de routage pour éviter la liaison qui est en panne)

Désavantage

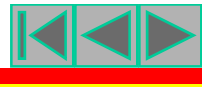
- Plus complexe à implémenter que routage statique (il va falloir définir des algorithmes distribués pour permettre aux routeurs de mettre à jours leurs tables de routage)

Méthodes

1. Routage avec vecteurs de distance
2. Routage avec état de liaison

(plus complexe à mettre en œuvre mais apporte des facilités supplémentaires)

Routage avec vecteurs de distance



Principe de base :

I. Chaque routeur transmet (à ses voisins) périodiquement un vecteur comprenant pour chaque destination connue du routeur une information indiquant :

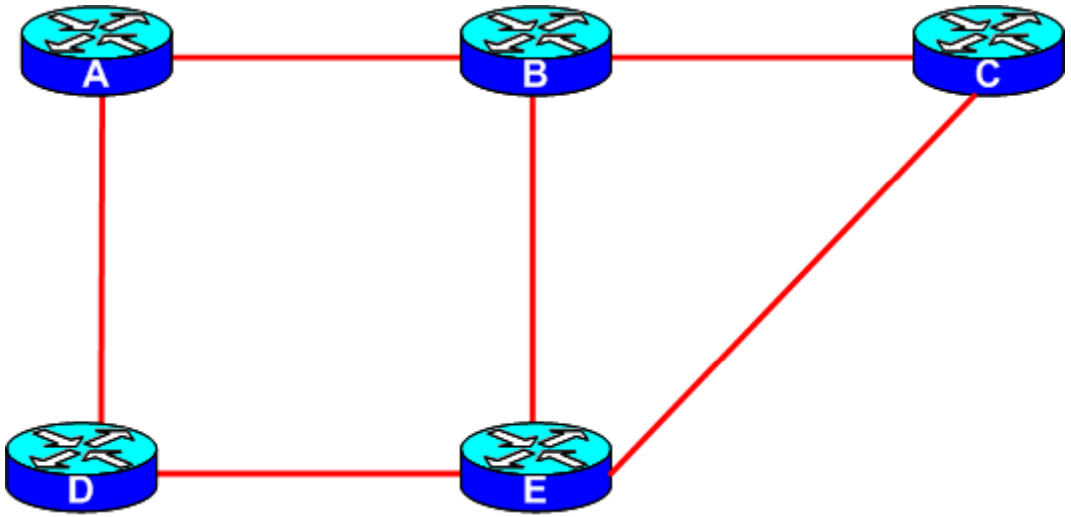
   1. Adresse de la destination
   2. distance depuis le routeur qui transmet le vecteur jusqu'à la destination
      - *« le vecteur de distance est en fait un résumé de la table de routage du routeur »*

ii. Chaque routeur

   1. reçoit les vecteurs transmis par ses voisins immédiats
   2. et se base sur cette information pour construire sa table de routage

iii. Lorsqu'il démarre, un routeur ne connaît que lui-même

- Exemple : Réseau simple avec lignes de distance unitaire
- Politique : On va choisir comme politique de routage celle qui minimise le nombre de routeurs intermédiaires (⇔ la route qui contient le minimum de nombre de routeurs intermédiaires)



- On considère qu'on vient d'installer le réseau, on a configuré chaque routeur avec son adresse et a démarré.

Exemple : Etat Initial

- On considère qu'on vient d'installer le réseau, on a configuré chaque routeur avec son adresse et a démarré.

Table de Routage
A : 0 (Local)

Table de Routage
B : 0 (Local)

Table de Routage
C: 0 (Local)

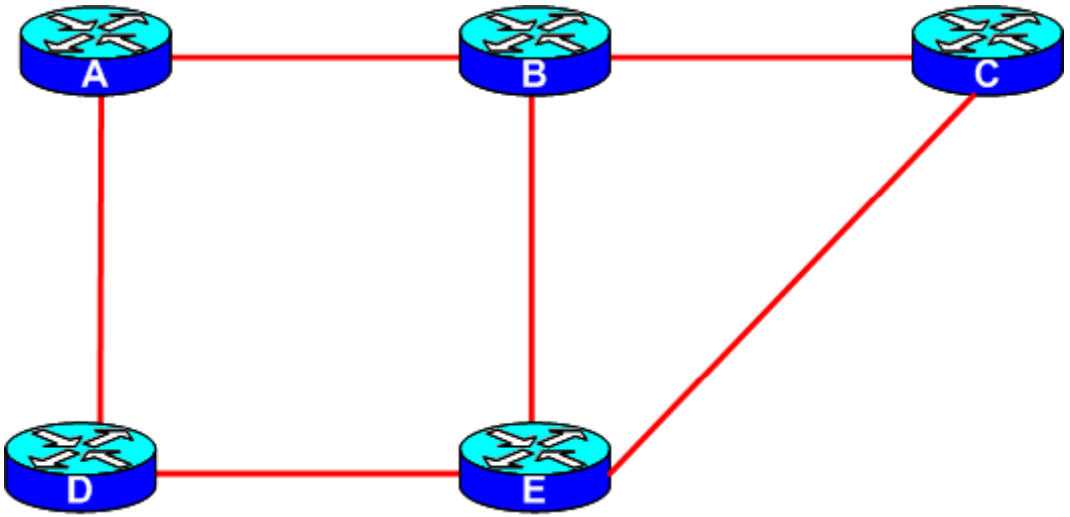


Table de Routage
D : 0 (Local)

Table de Routage
E : 0 (Local)

Vecteurs de Distance - Etape 1.0

Table de Routage
A : 0 (Local)

Table de Routage
B : 0 (Local)

Table de Routage
C: 0 (Local)

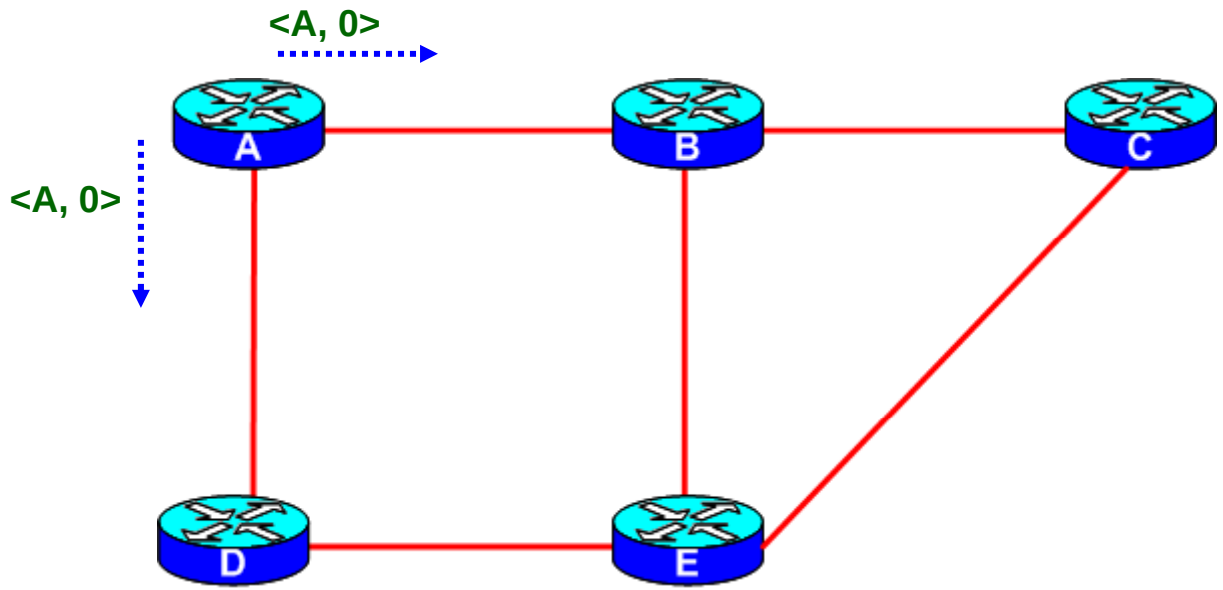


Table de Routage
D : 0 (Local)

Table de Routage
E : 0 (Local)

Vecteurs de Distance - Etape 1.1

Table de Routage
A : 0 (Local)

Table de Routage
B : 0 (Local) A : 1 (Ouest)

Table de Routage
C: 0 (Local)

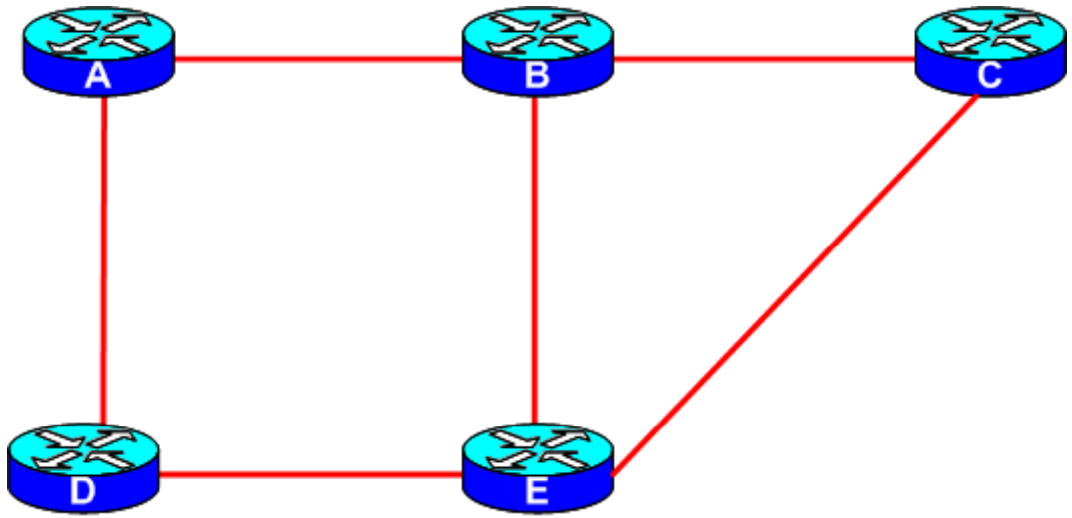


Table de Routage
D : 0 (Local) A : 1 (Nord)

Table de Routage
E : 0 (Local)

Vecteurs de Distance - Etape 2.0

Table de Routage
A : 0 (Local)

Table de Routage
B : 0 (Local) A : 1 (Ouest)

Table de Routage
C: 0 (Local)

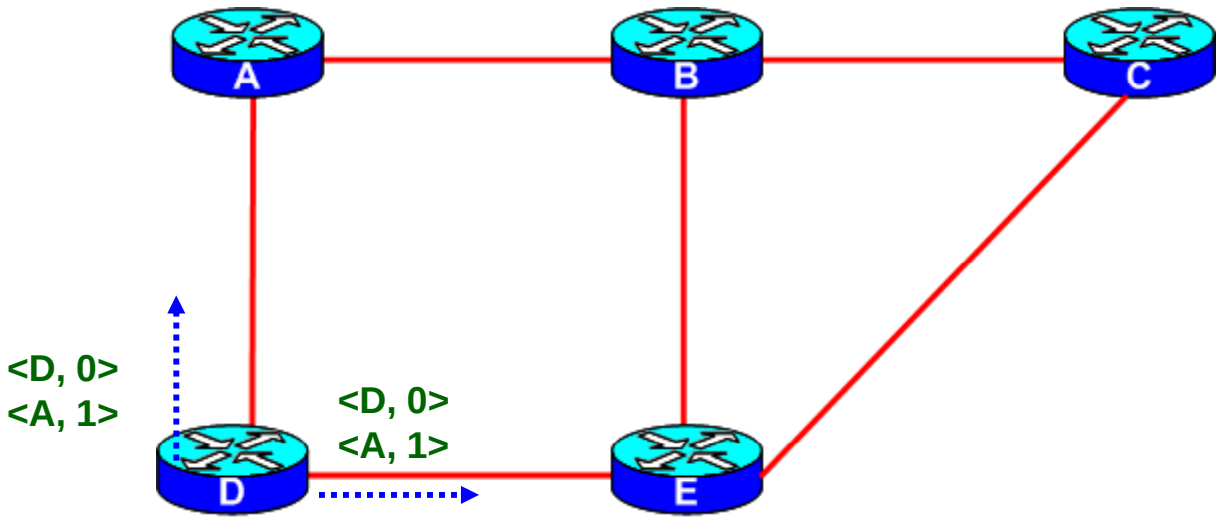
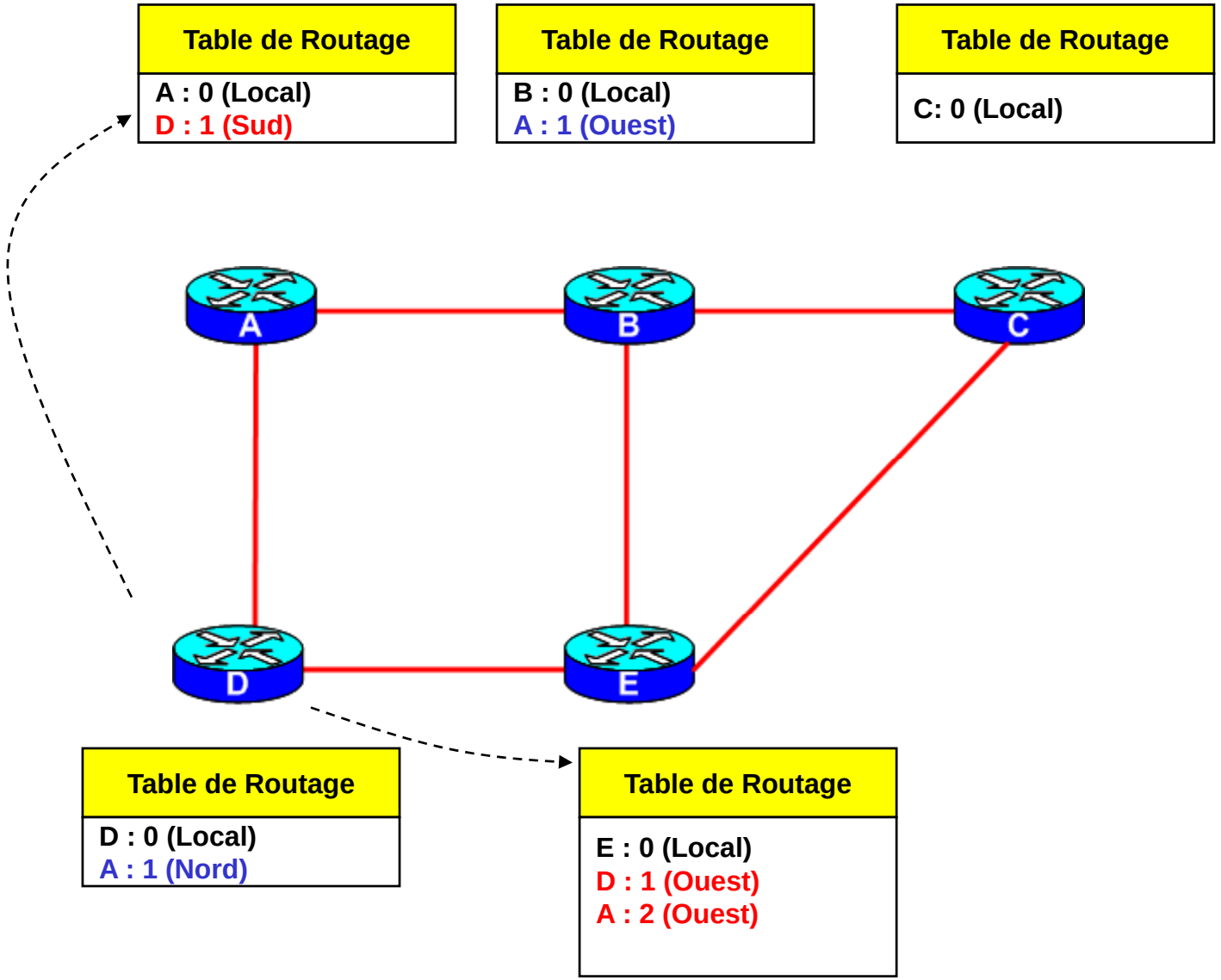


Table de Routage
D : 0 (Local) A : 1 (Nord)

Table de Routage
E : 0 (Local)

Vecteurs de Distance - Etape 2.1



Vecteurs de Distance - Etape 3.0

Table de Routage
A : 0 (Local) D : 1 (Sud)

Table de Routage
B : 0 (Local) A : 1 (Ouest)

Table de Routage
C: 0 (Local)

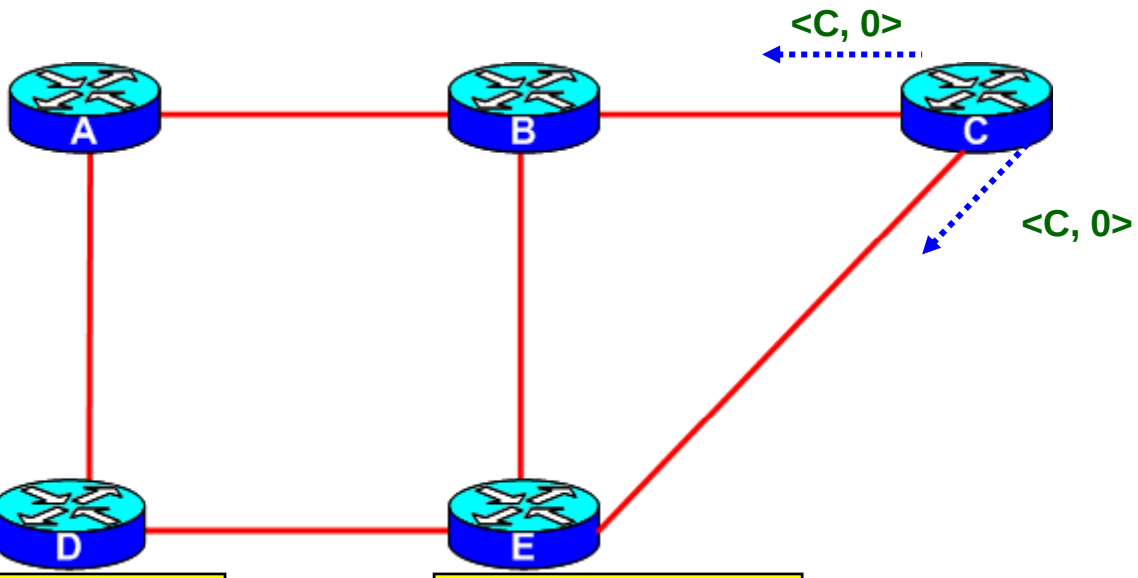
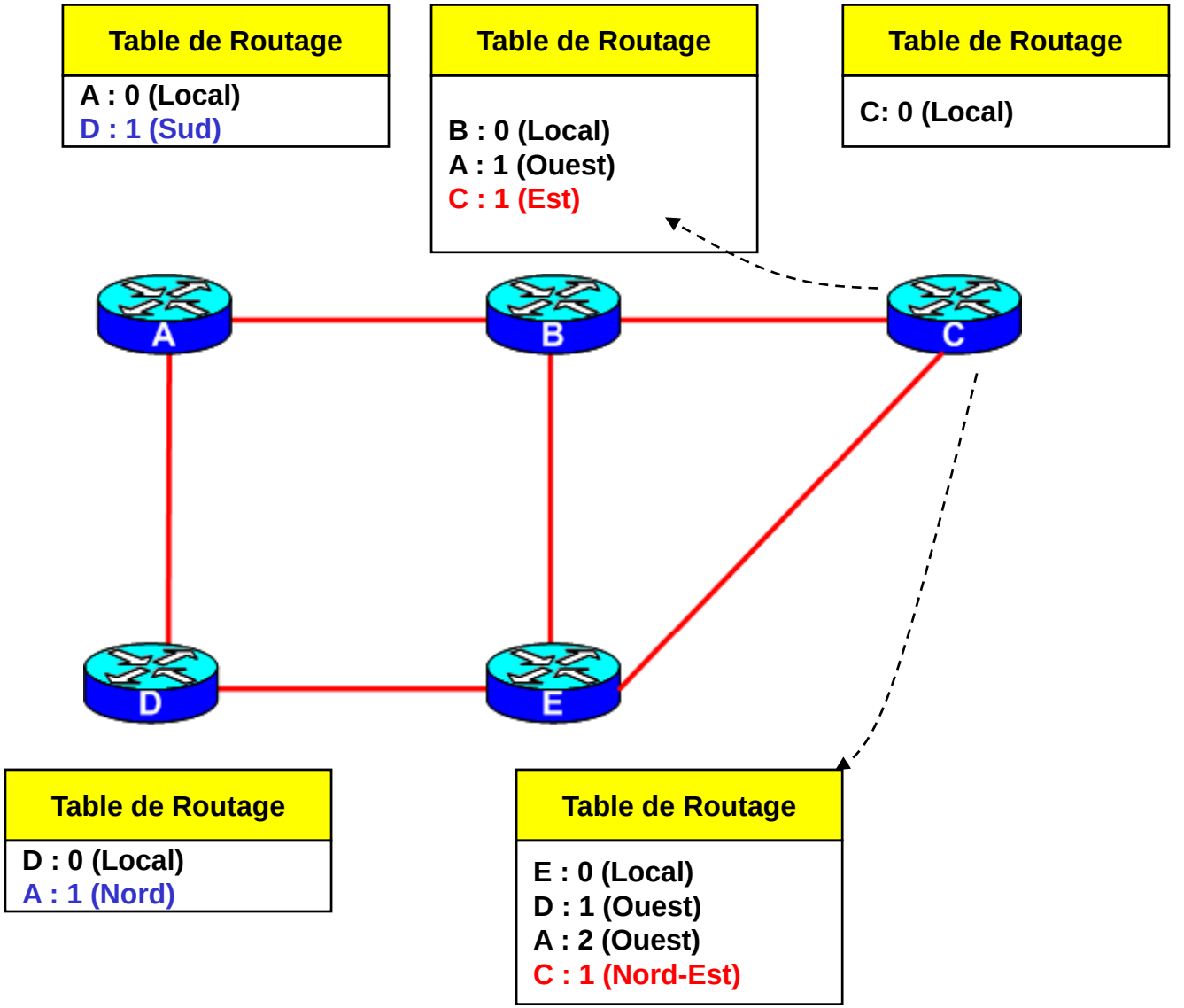


Table de Routage
D : 0 (Local) A : 1 (Nord)

Table de Routage
E : 0 (Local) D : 1 (Ouest) A : 2 (Ouest)

Ces vecteurs de distance vont forcer une MAJ des tables de routages des deux routeurs B et E.



Supposons maintenant que Le routeur E va envoyer son vecteur de distance

Table de Routage
A : 0 (Local) D : 1 (Sud)

Table de Routage
B : 0 (Local) A : 1 (Ouest) C : 1 (Est)

Table de Routage
C: 0 (Local)

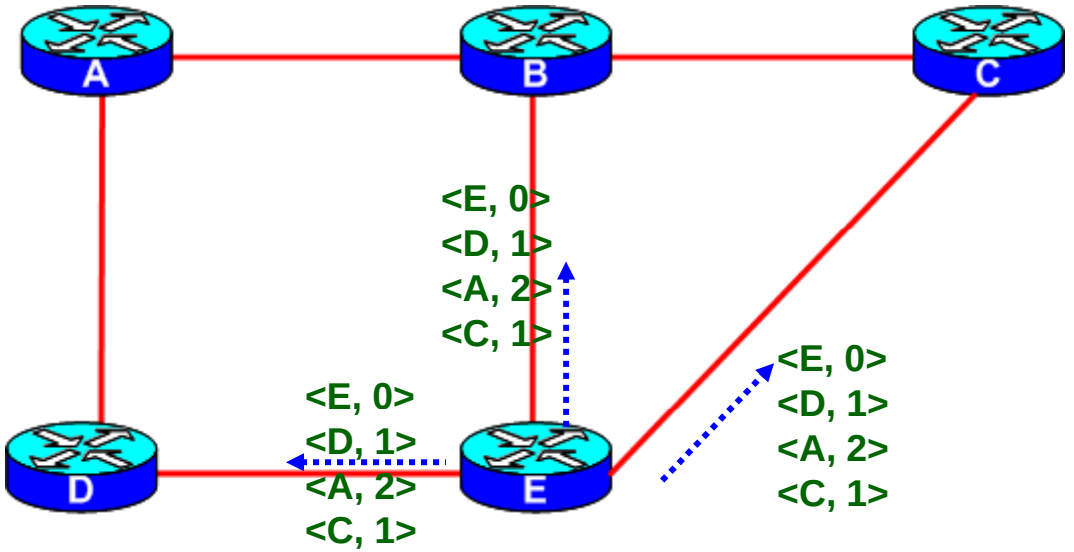


Table de Routage
D : 0 (Local) A : 1 (Nord)

Table de Routage
E : 0 (Local) D : 1 (Ouest) A : 2 (Ouest) C : 1 (Nord-Est)

Réception de ce vecteur par B ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C: 0 (Local)

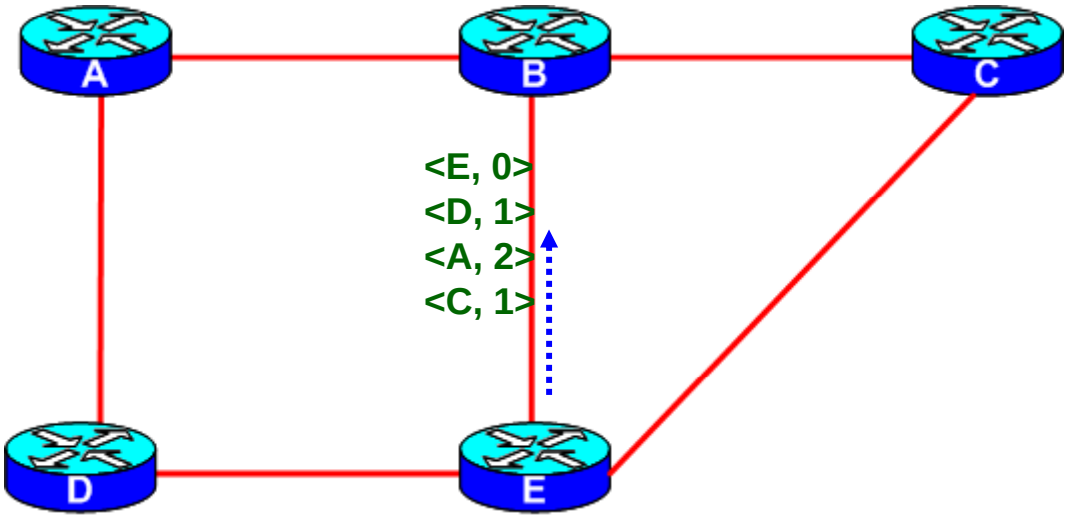


Table de Routage
D : 0 (Local)
A : 1 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)

Réception de ce vecteur par D,

Table de Routage
A : 0 (Local) D : 1 (Sud)

Table de Routage
B : 0 (Local) A : 1 (Ouest) C : 1 (Est) D : 2 (Sud) E : 1 (Sud)

Table de Routage
C: 0 (Local)

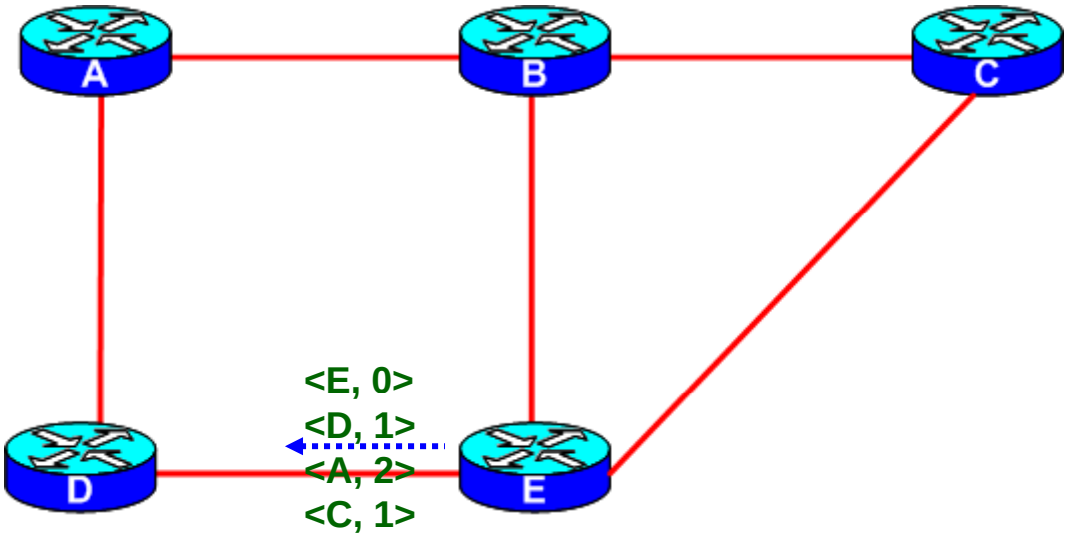


Table de Routage
D : 0 (Local) A : 1 (Nord) E : 1 (Est) C : 2 (Est)

Table de Routage
E : 0 (Local) D : 1 (Ouest) A : 2 (Ouest) C : 1 (Nord-Est)

Réception de ce vecteur par C ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 3(Sud-Ouest)

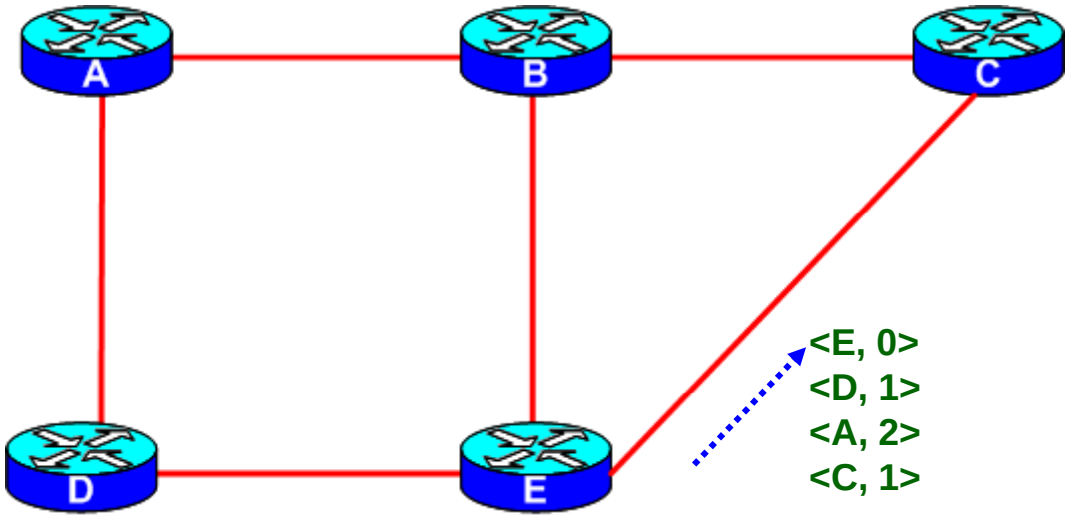


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)

Supposons maintenant que Le routeur B va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 3(Sud-Ouest)

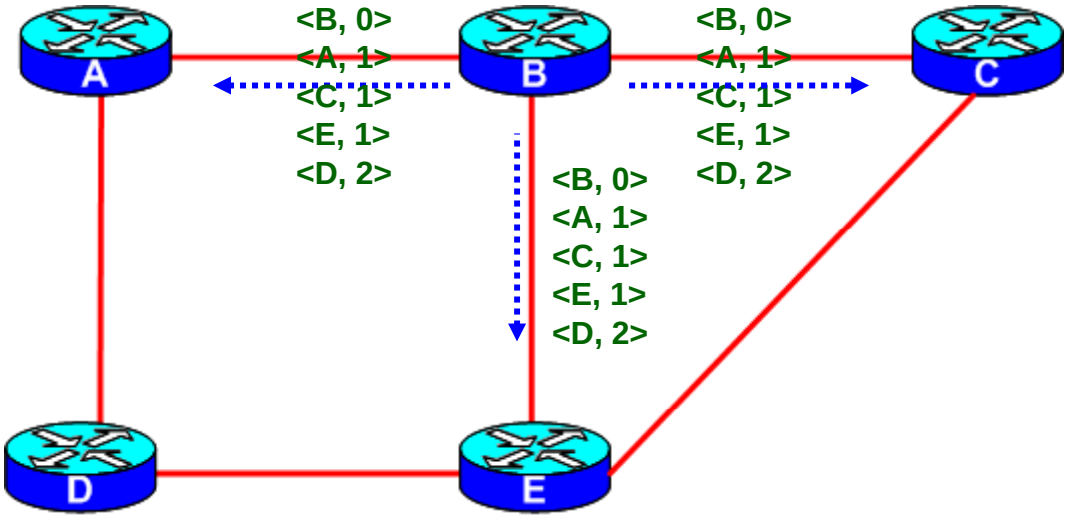


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)

Réception de ce vecteur par A ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 1 (Est)
C : 2 (Est)
E : 2 (Est)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 3 (Sud-Ouest)

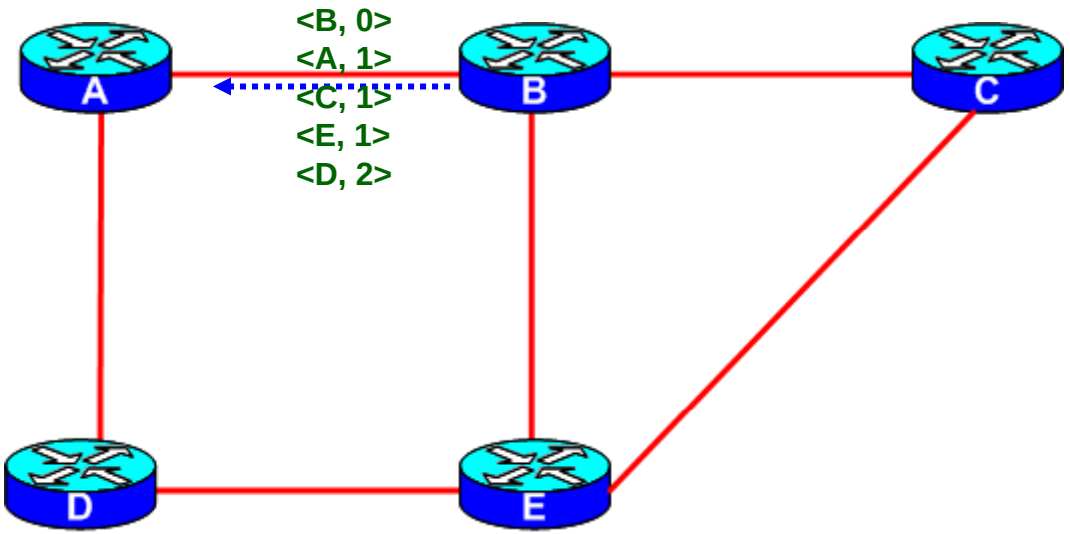


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)

Réception de ce vecteur par C ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 1 (Est)
C : 2 (Est)
E : 2 (Est)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

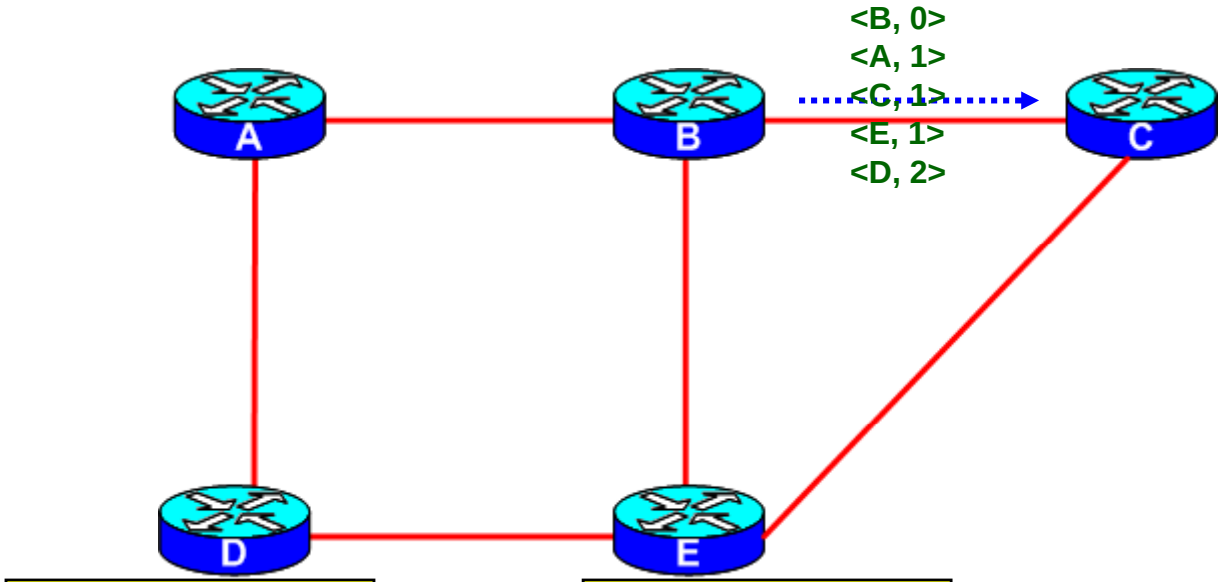


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)

Réception de ce vecteur par E ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 1 (Est)
C : 2 (Est)
E : 2 (Est)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

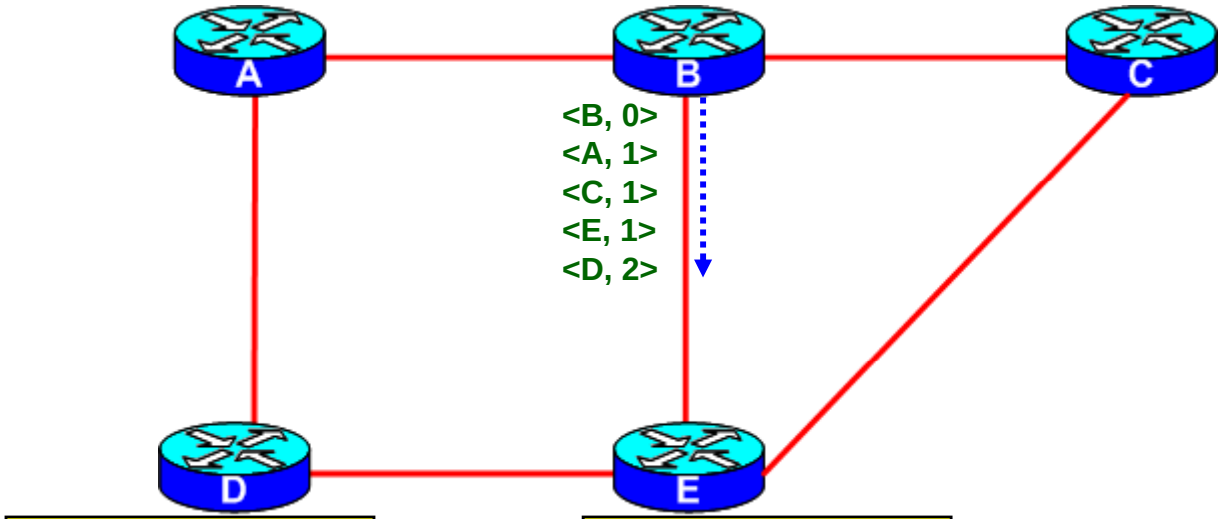


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Supposons maintenant que Le routeur A va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 1 (Est)
C : 2 (Est)
E : 2 (Est)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

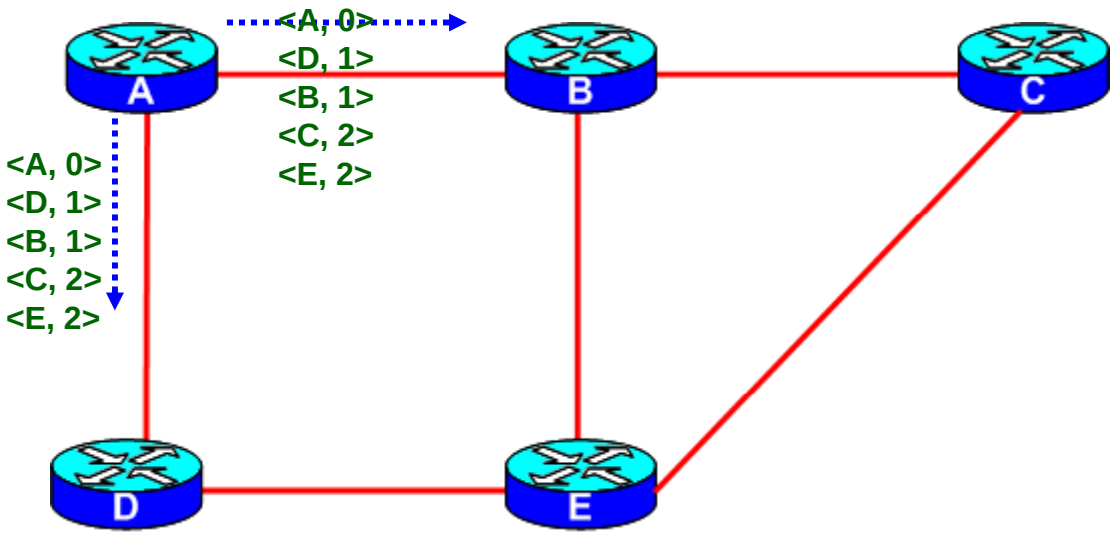


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par B ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 1 (Est)
C : 2 (Est)
E : 2 (Est)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

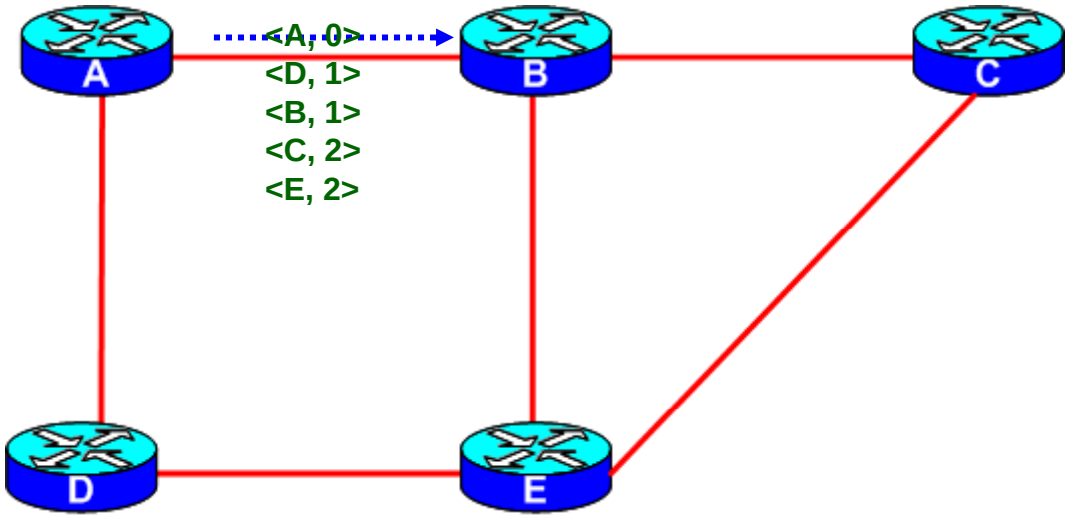


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par D ,

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 1 (Est)
C : 2 (Est)
E : 2 (Est)

Table de Routage
B : 0 (Local)
A : 1 (Ouest)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

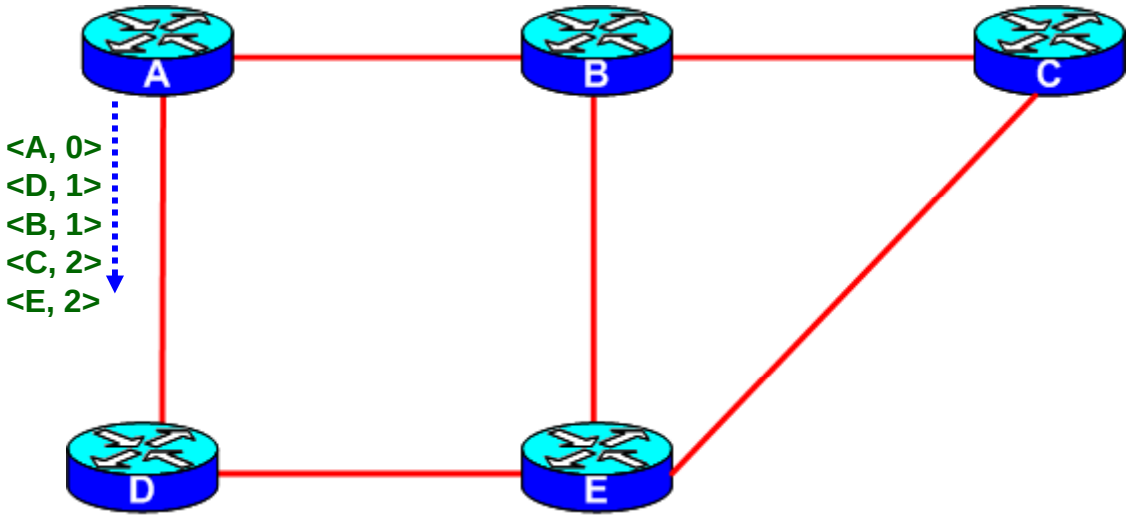
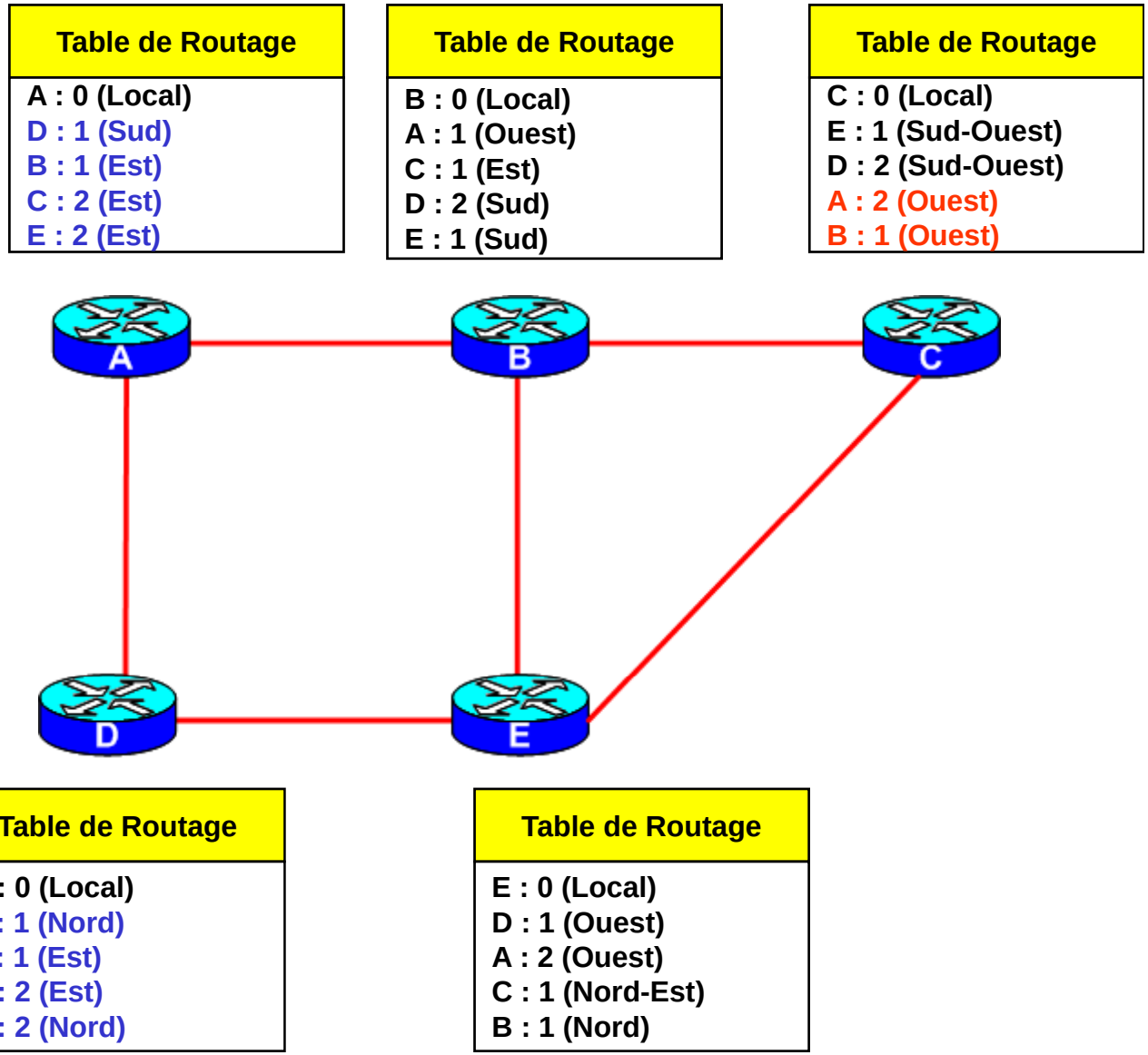


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Tables de routage définitives

- A cet instant, tous les routeurs savent comment joindre n'importe quel autre routeur du réseau
- les tables de routage sont stables
  - la réémission d'un vecteur de distance ne provoquera pas de changement dans une table de routage



Les vecteurs de distance sont envoyés périodiquement toutes les trente secondes.

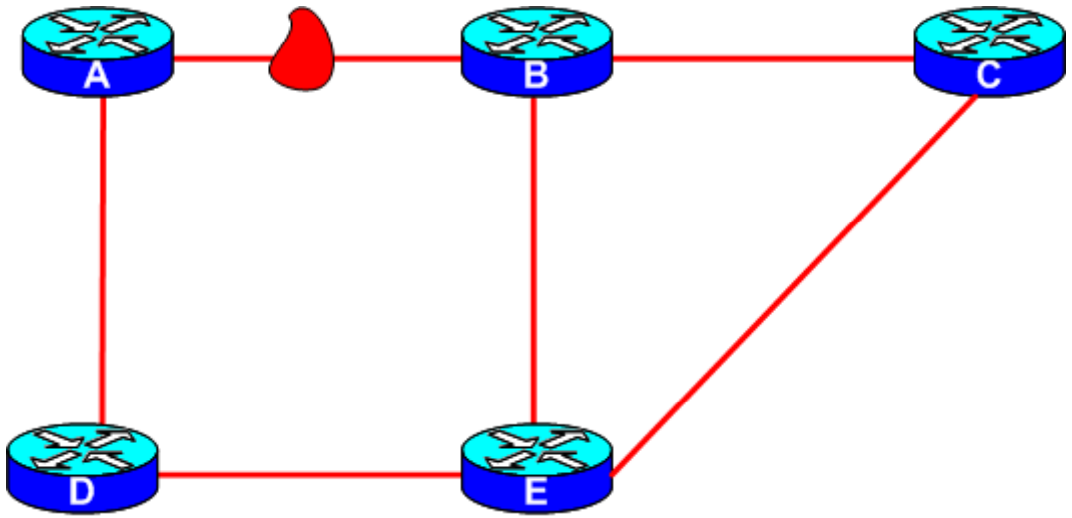
vecteurs de distance
- Réaction en cas de pannes -

- Comment Réagir :

1. Supprimer dans la table de routage de A et B les destinations que l'on pouvait joindre via la ligne en panne
2. Ceci ne suffit pas, il faut également propager dans le réseau l'information concernant la panne de la ligne

Idées :

- ✓ une ligne en panne = ligne avec distance infinie
- ✓ envoyer un nouveau vecteur de distance indiquant une distance infinie pour les destinations que l'on pouvait joindre via la ligne qui est tombée en panne



Cas de panne d'une ligne 1

Le routeur A va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : ∞
E : ∞

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Ouest)
D : 2 (Sud- Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

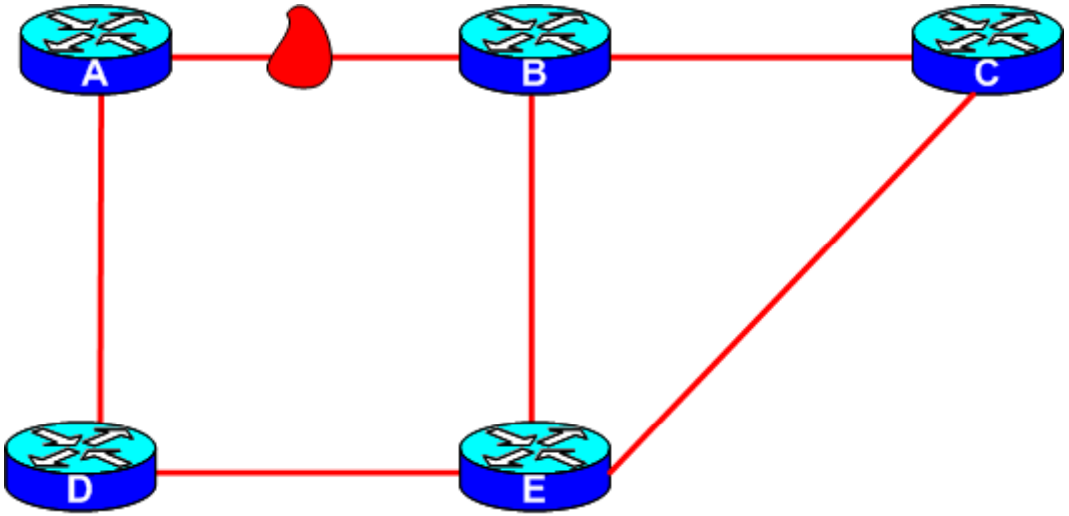


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par D

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : ∞
E : ∞

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
D : 2 (Sud- Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

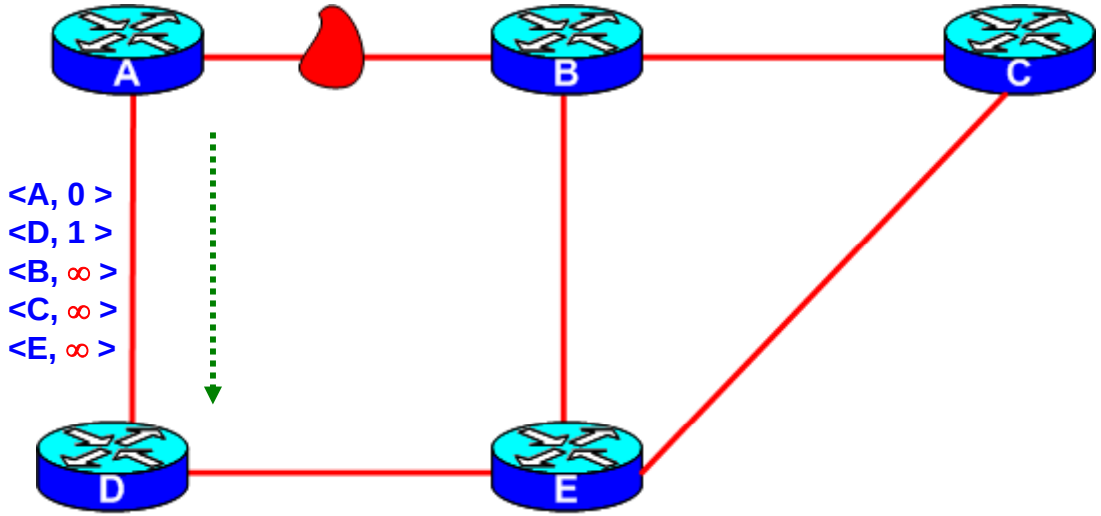


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Le routeur D va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : ∞
E : ∞

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
D : 2 (Sud- Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

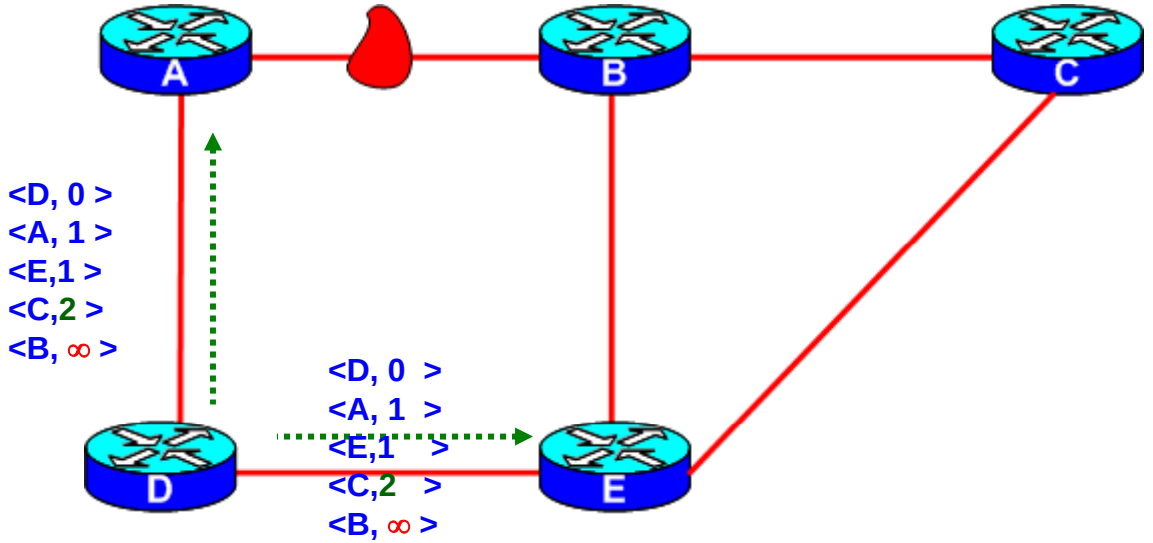


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : ∞

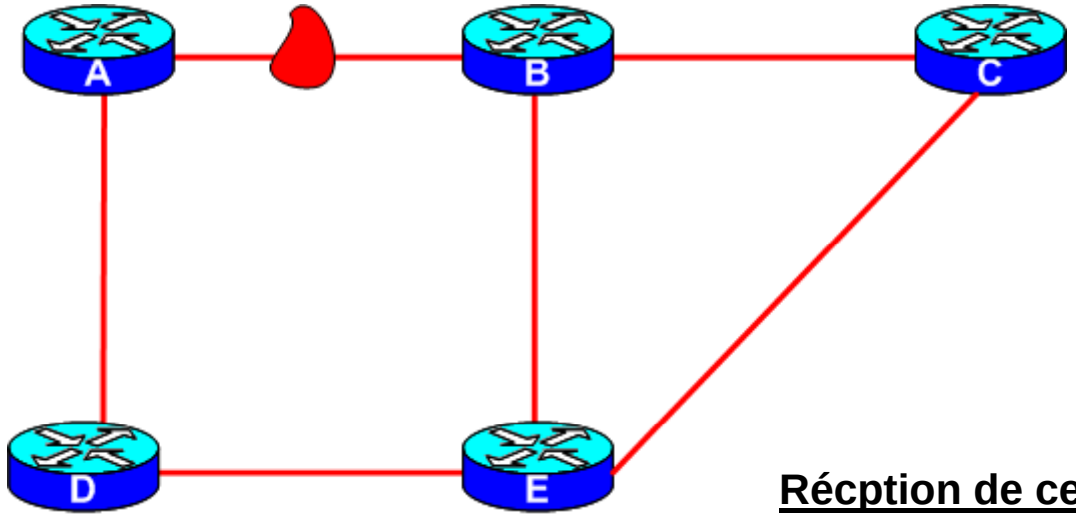
Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par A

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest t)
D : 2 (Sud- Ouest)
A : 2 (Ouest)
B : 1 (Ouest)



Récption de ce vecteur par E

Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Le routeur B va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
D : 2 (Sud- Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

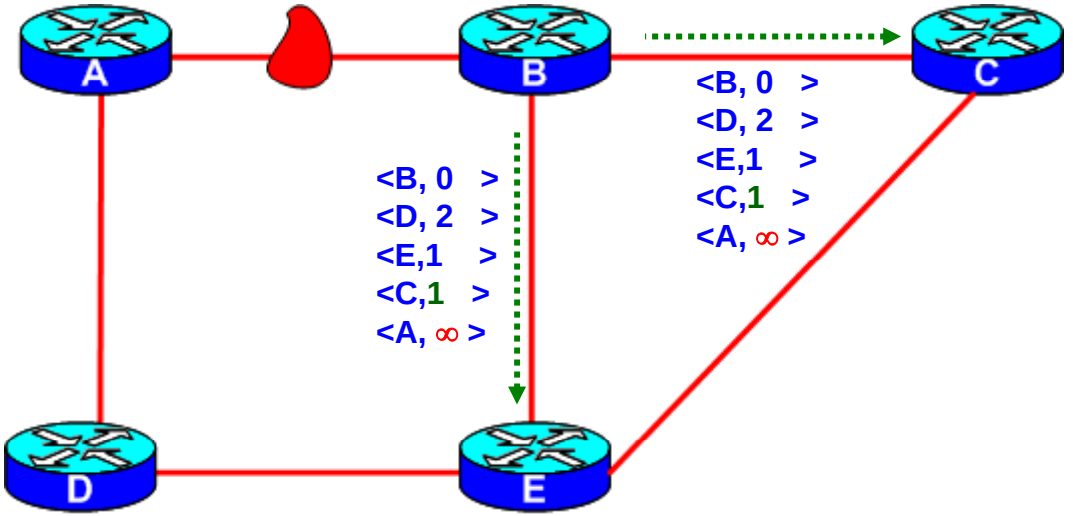


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par E

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest t)
D : 2 (Sud- Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

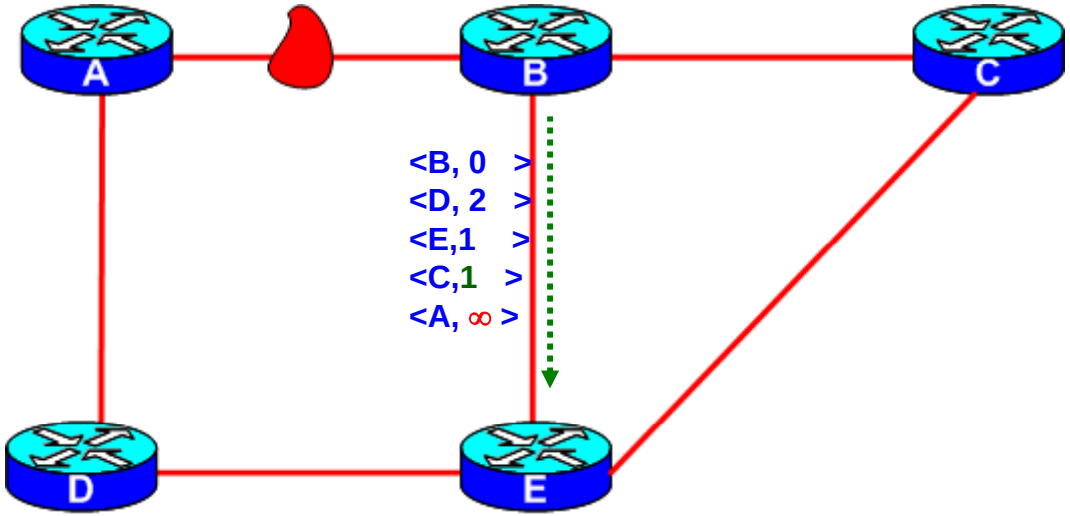


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par C

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
D : 2 (Sud- Ouest)
A : ∞
B : 1 (Ouest)

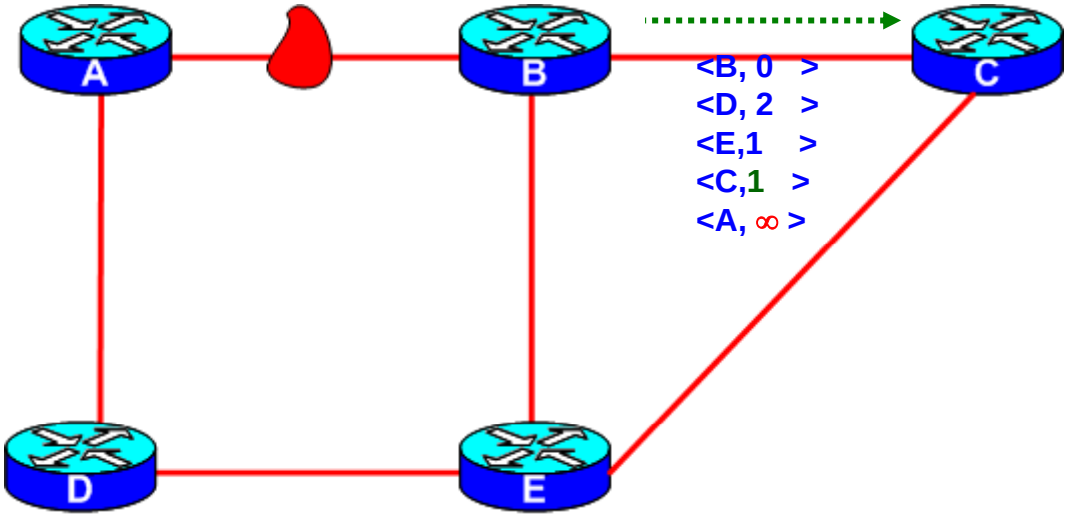


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Le routeur E va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : ∞
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : ∞
D : 2 (Sud- Ouest)
B : 1 (Ouest)

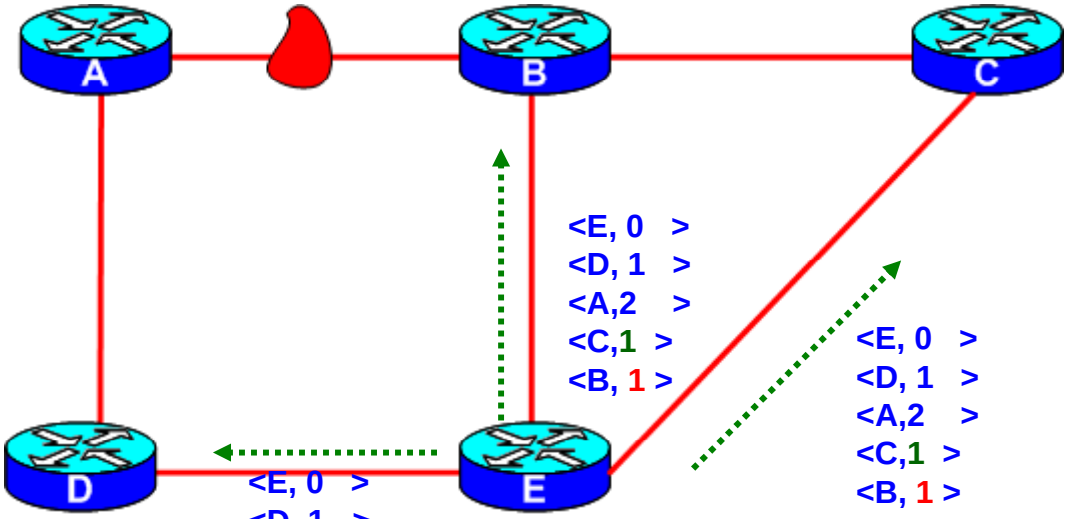


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par D

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

Réception de ce vecteur par B

Réception de ce vecteur par C

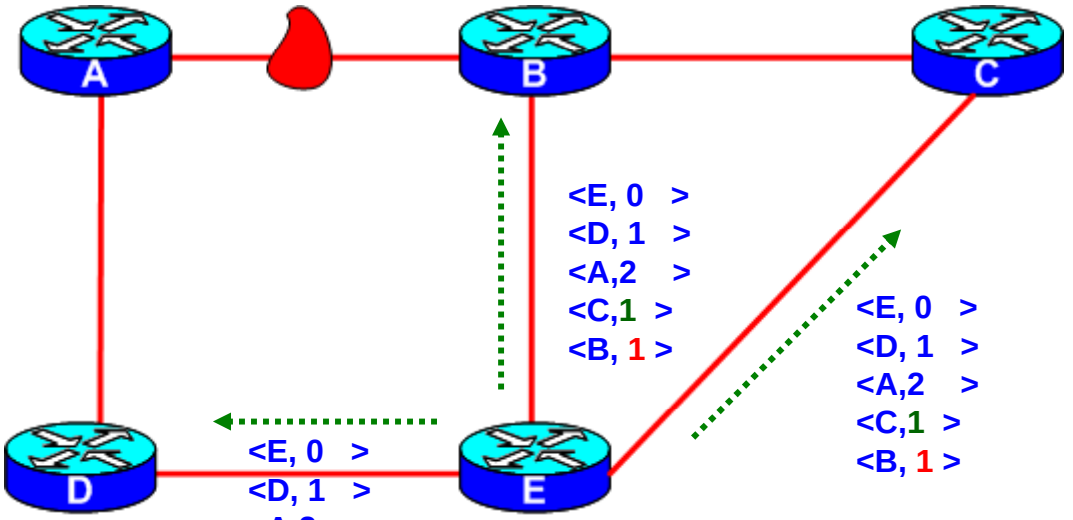


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Le routeur D va envoyer son vecteur de distance

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : ∞
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest t)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

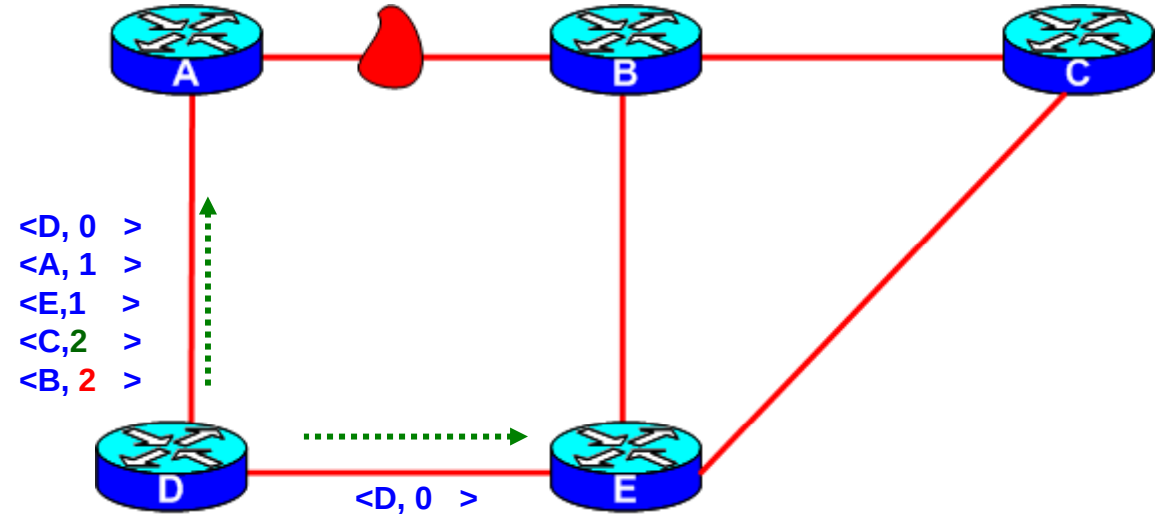


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Est)

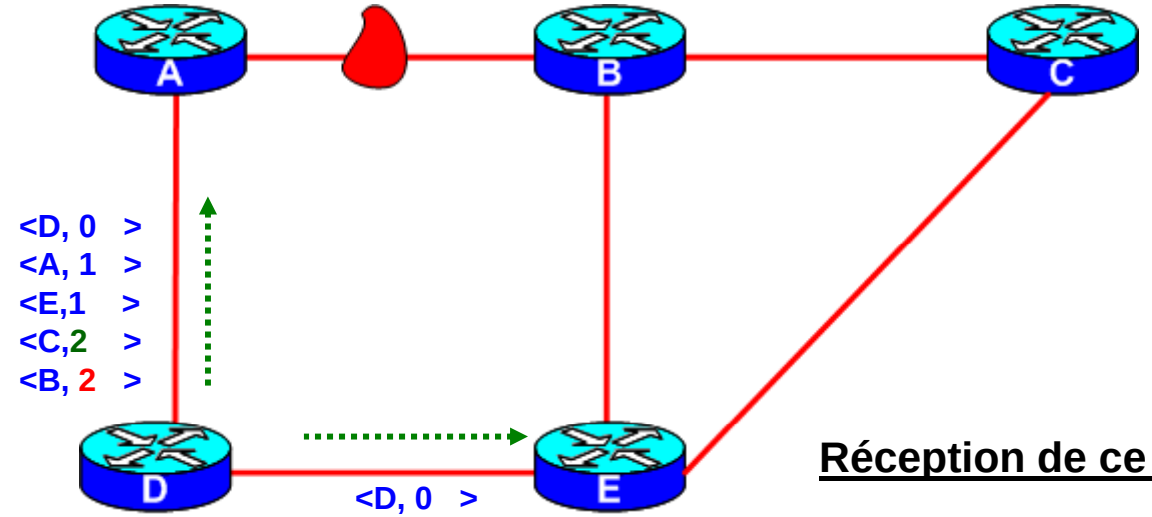
Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Réception de ce vecteur par A

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest t)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)



Réception de ce vecteur par E

Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Nouvelles tables de routage

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

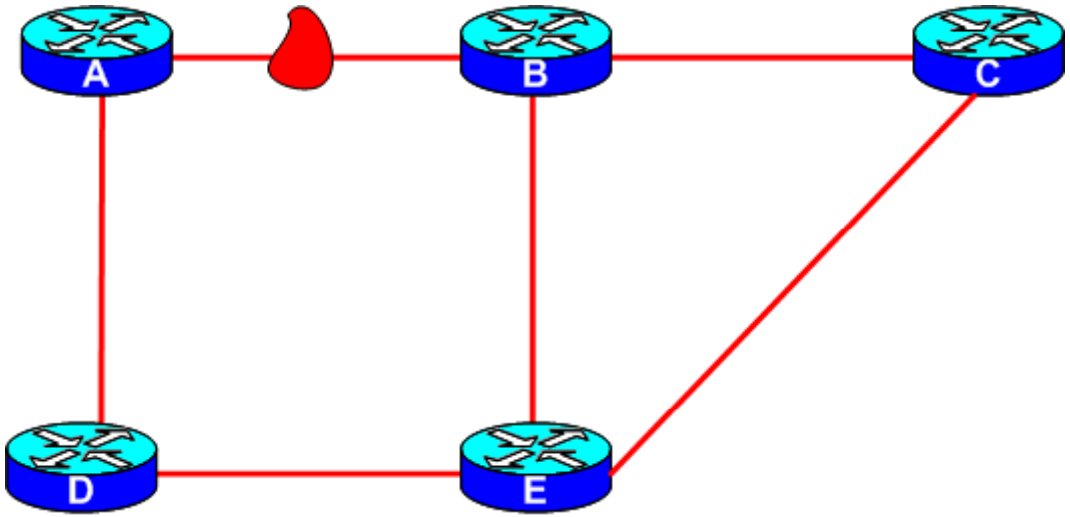


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Vecteurs de Distance
Nouvelle Panne

Nouvelle Panne sur la ligne entre D et E

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

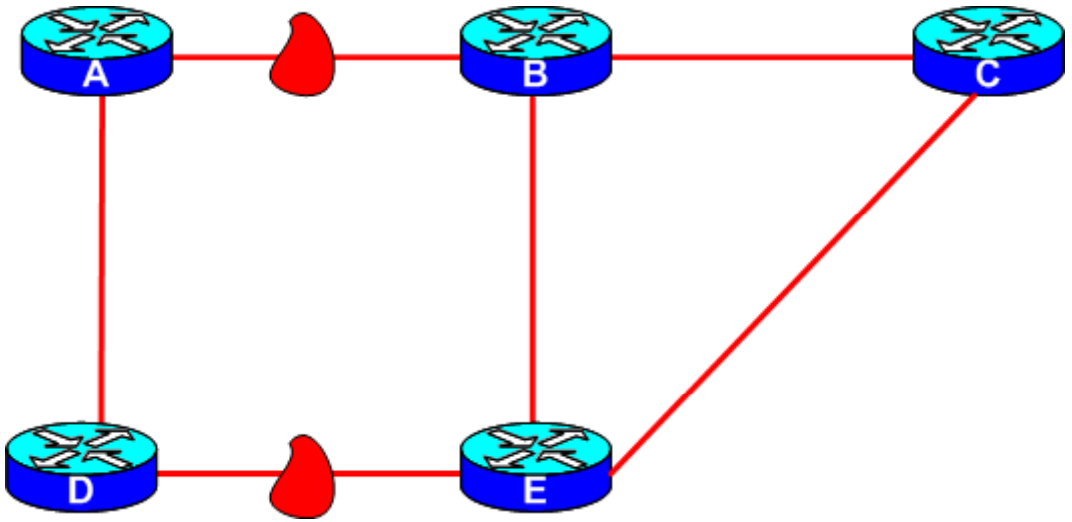


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Cas d'une nouvelle panne - 1

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

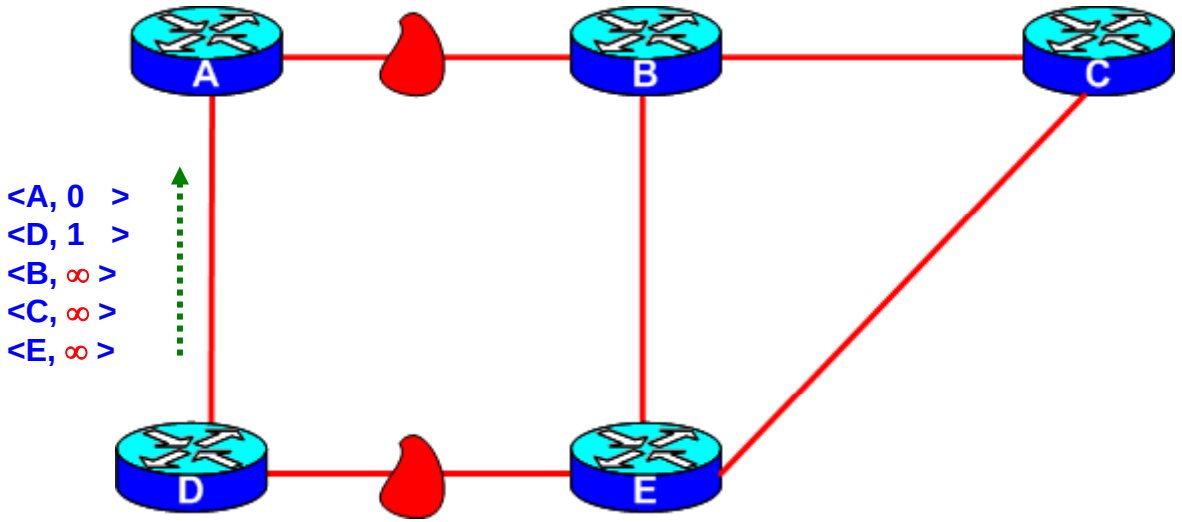


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : ∞
C : ∞
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

- Si **D** envoie son vecteur de distance, pas de problème
- **A** sera au courant qu'il ne peut pas joindre B,C et E via son **port sud**

Cas d'une nouvelle panne - 2

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

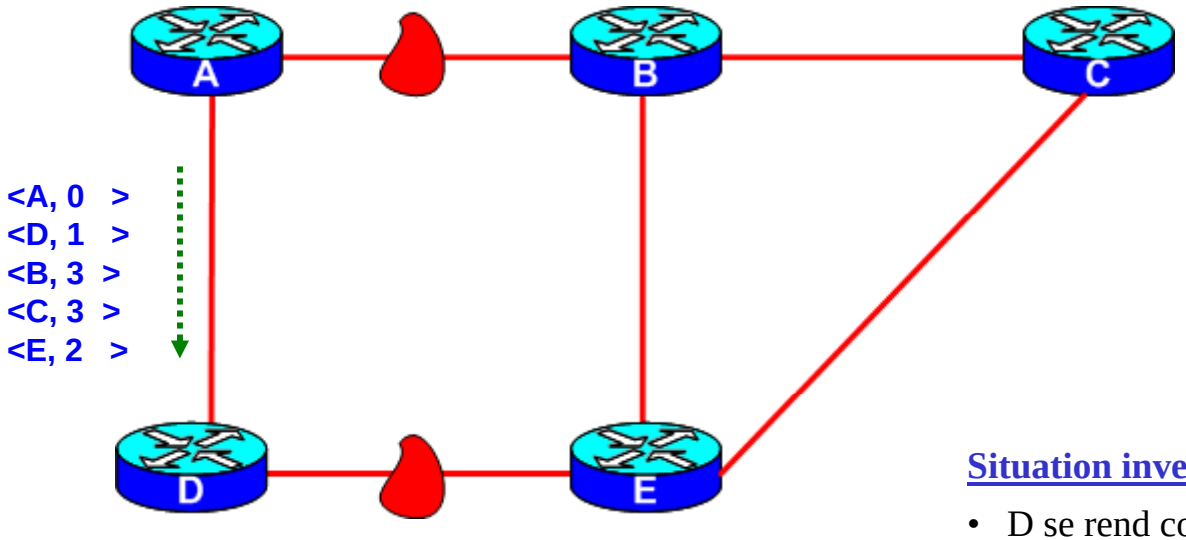


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : ∞
C : ∞
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Situation inverse :

- D se rend compte de la panne => E est à ∞, B est à ∞ , C est à ∞
- A envoie son vecteur avant D
- Vecteur de distance de A

A=0; D=1; B=3; C=3; E=2

envoyé sur tous les ports de A avant que D envoi son vecteur

Cas d'une nouvelle panne - 3

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

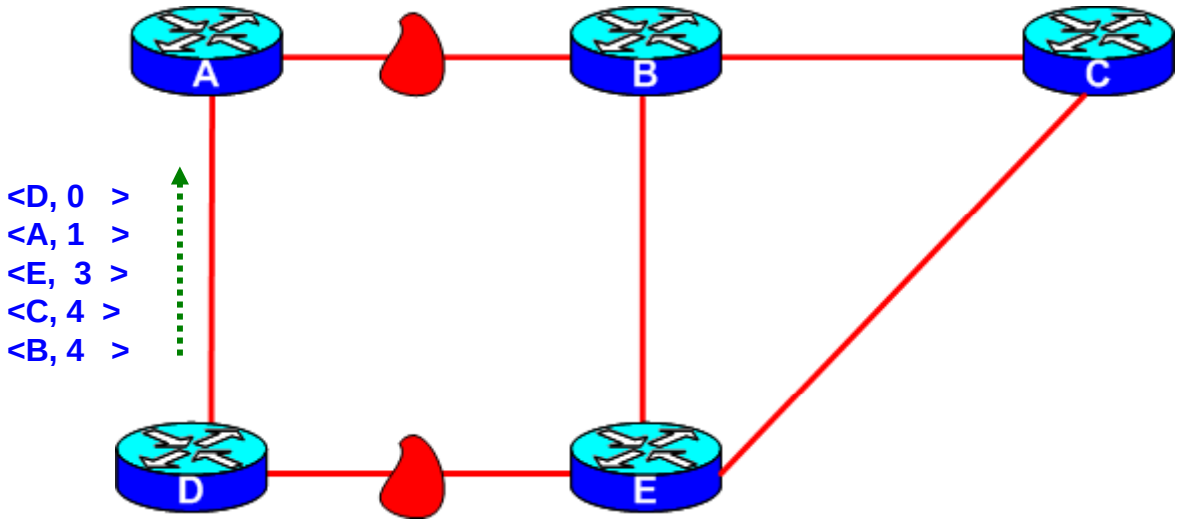


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 3 (Nord)
C : 4 (Nord)
B : 4 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Cas d'une nouvelle panne - 4

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 4 (Sud)
C : 4(Sud)
E : 3 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

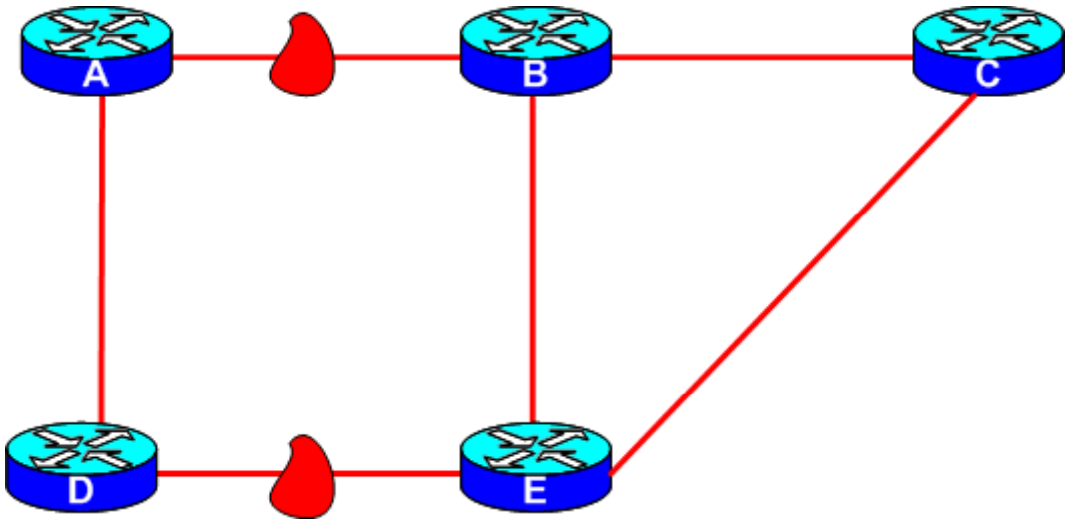


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 3 (Nord)
C : 4 (Nord)
B : 4 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Cas d'une nouvelle panne - 5

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 4 (Sud)
C : 4(Sud)
E : 3 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

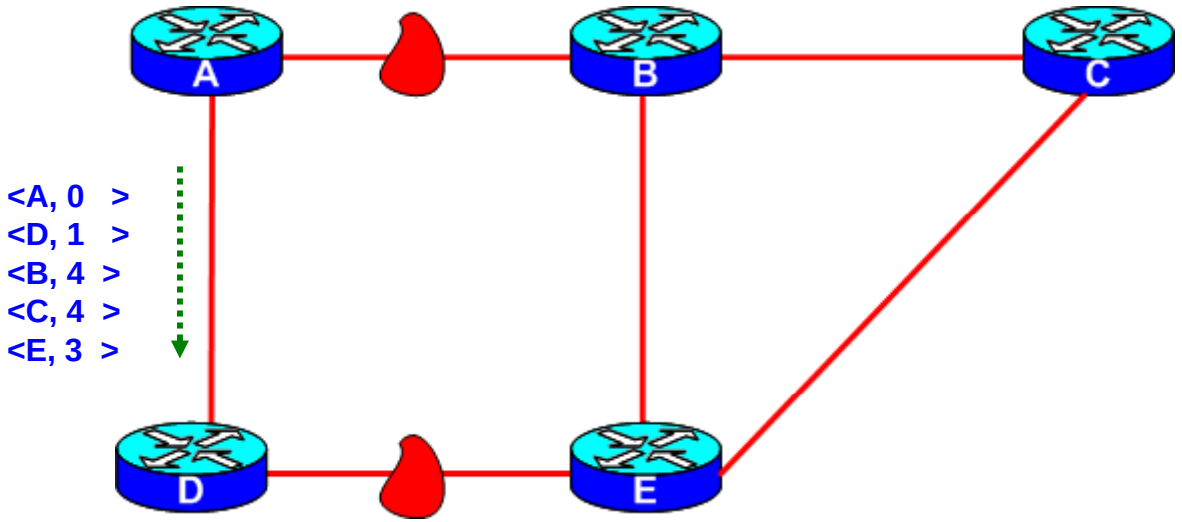


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 3 (Nord)
C : 4 (Nord)
B : 4 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Cas d'une nouvelle panne - 6

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 4 (Sud)
C : 4(Sud)
E : 3 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

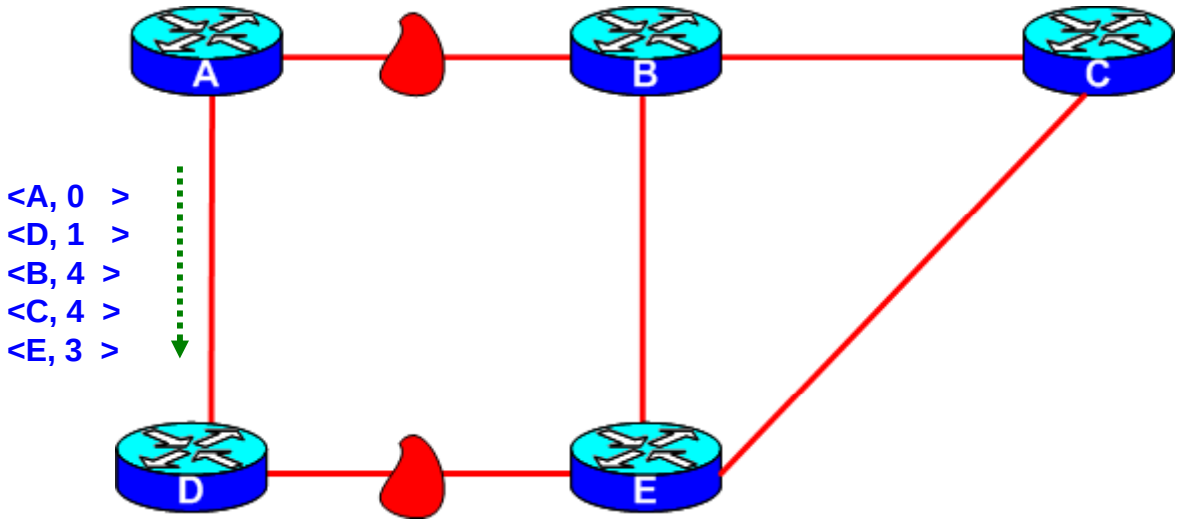


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 4 (Nord)
C : 5 (Nord)
B : 5 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Cas d'une nouvelle panne - 7

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 4 (Sud)
C : 4(Sud)
E : 3 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud- Ouest)
A : 3(Sud-Ouest)
D : 2 (Sud- Ouest)
B : 1 (Ouest)

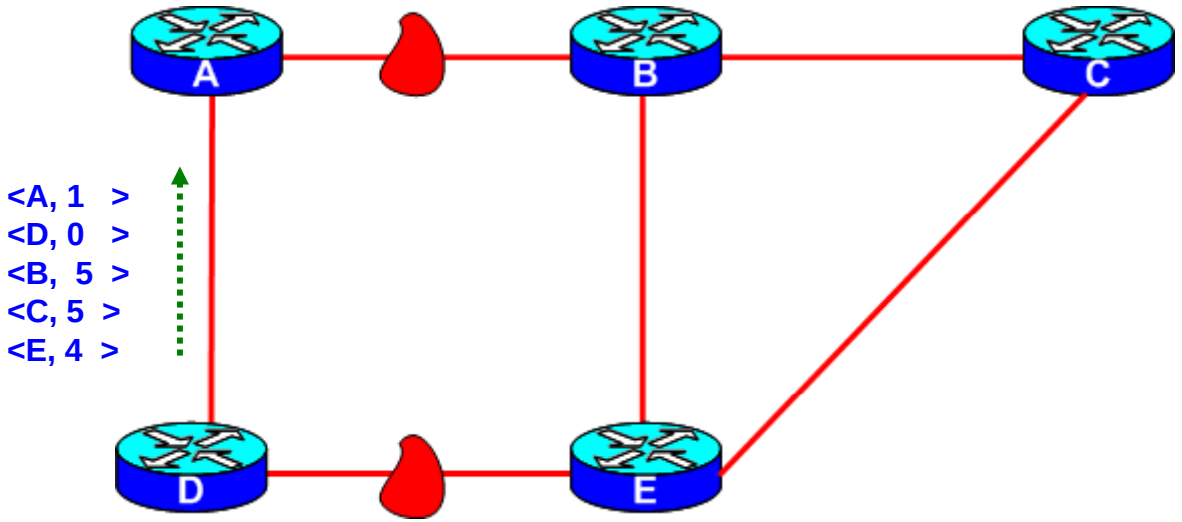
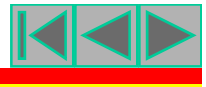


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 4 (Nord)
C : 5 (Nord)
B : 5 (Nord)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)



- Origine du problème de comptage à l' ∞ :
  - Un routeur annonce sur une ligne des routes qu'il a apprises via cette même ligne
- Comment éviter le comptage à l'infini ?

  → horizon partagé (split horizon)
- principe

  1. construire un vecteur de distance pour chaque ligne
  2. sur une ligne, ne *pas* annoncer les destinations que l'on parvient à joindre via cette ligne

Horizon partagé

Horizon partagé –1-

- **Retour à l'exemple précédent**

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Est)
A : 3(Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

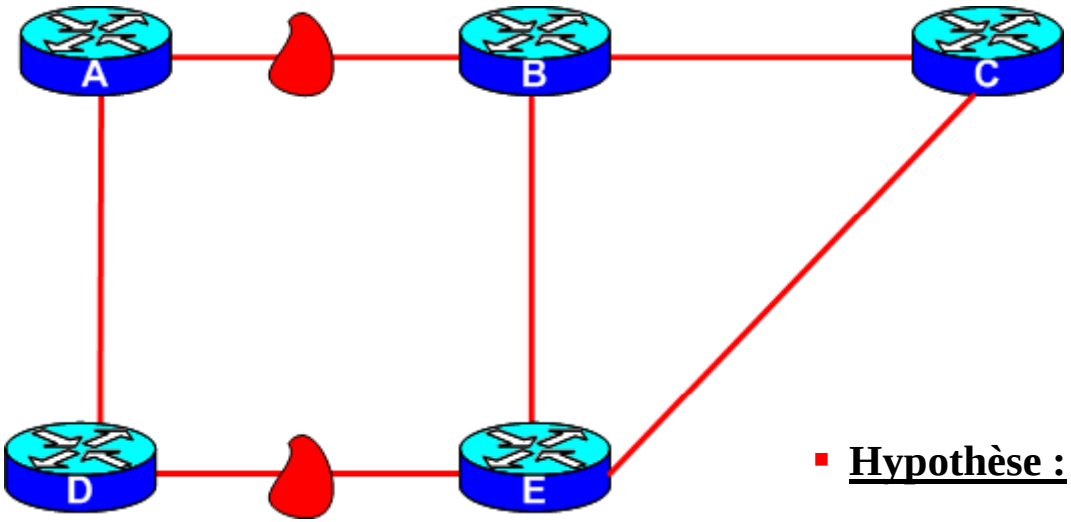


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : 1 (Est)
C : 2 (Est)
B : 2 (Est)

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

- **Hypothèse :**
  - ne pas annoncer sur son port sud les routeurs qu'il peut joindre **à partir de ce port**, on peut résoudre le problème de comptage à l'infinie

Horizon partagé – 2-

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Est)
A : 3(Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

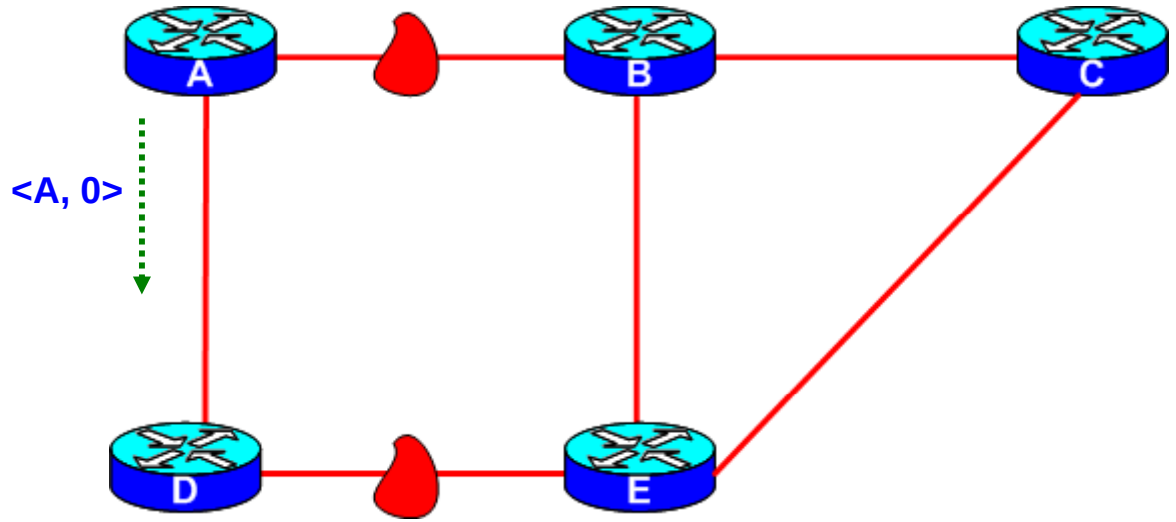


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : ∞
C : ∞
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Horizon partagé – 3-

Table de Routage
A : 0 (Local)
D : 1 (Sud)
B : 3 (Sud)
C : 3(Sud)
E : 2 (Sud)

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Est)
A : 3(Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

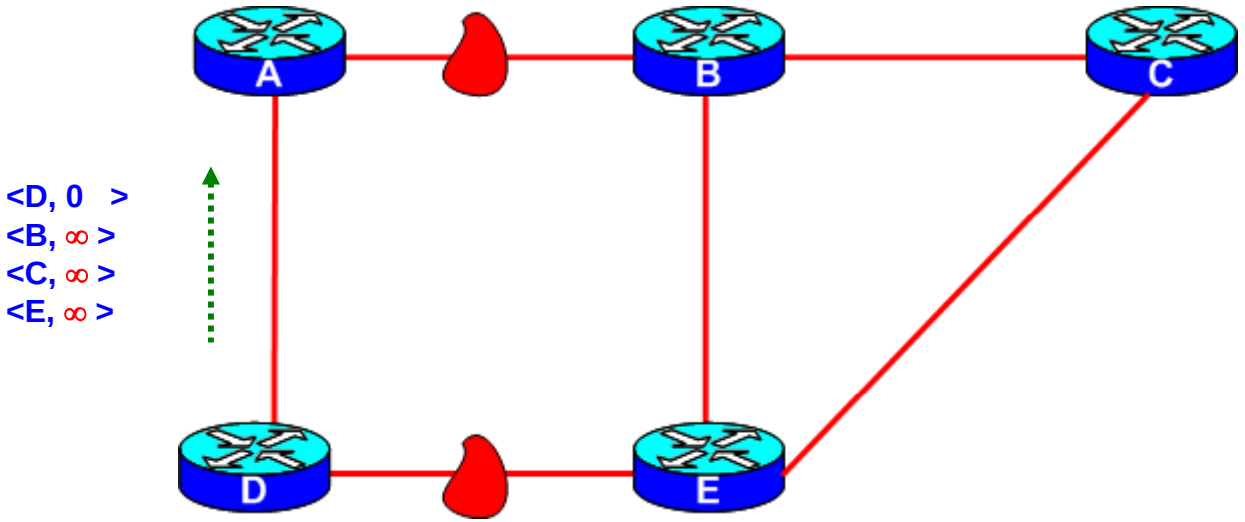


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : ∞
C : ∞
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)

Horizon partagé – 4-

Table de Routage
A : 0 (Local)
D : 1 (Sud)
<B, ∞ >
<C, ∞ >
<E, ∞ >

Table de Routage
B : 0 (Local)
A : 3(Sud)
C : 1 (Est)
D : 2 (Sud)
E : 1 (Sud)

Table de Routage
C : 0 (Local)
E : 1 (Sud-Est)
A : 3(Sud-Ouest)
A : 2 (Ouest)
B : 1 (Ouest)

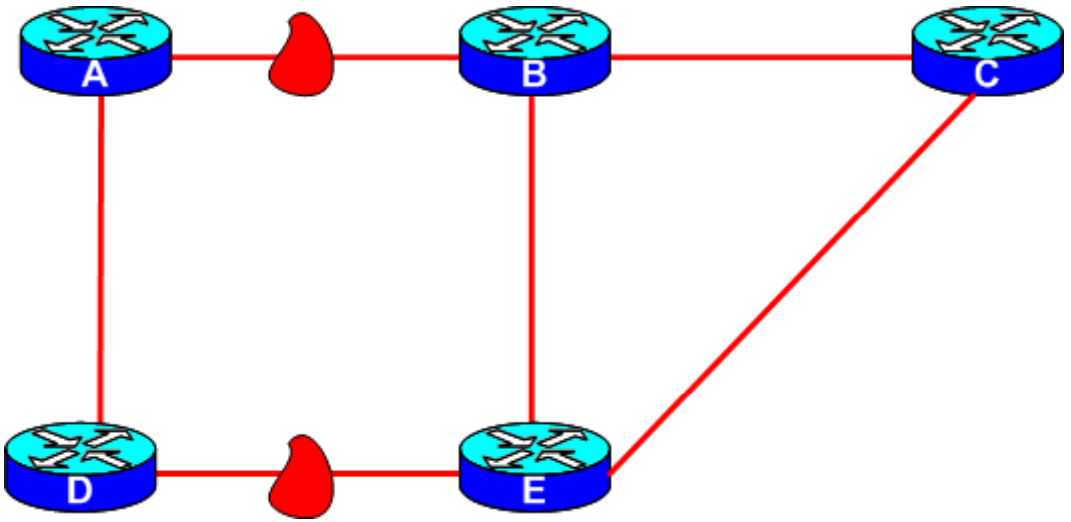
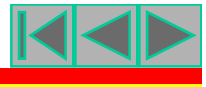


Table de Routage
D : 0 (Local)
A : 1 (Nord)
E : ∞
C : ∞
B : ∞

Table de Routage
E : 0 (Local)
D : 1 (Ouest)
A : 2 (Ouest)
C : 1 (Nord-Est)
B : 1 (Nord)



- Amélioration à horizon partagé :
  - horizon **partagé** avec **empoisonnement**
  - Principe
    - Construire un vecteur de distance pour chaque ligne
    - sur une ligne donnée :

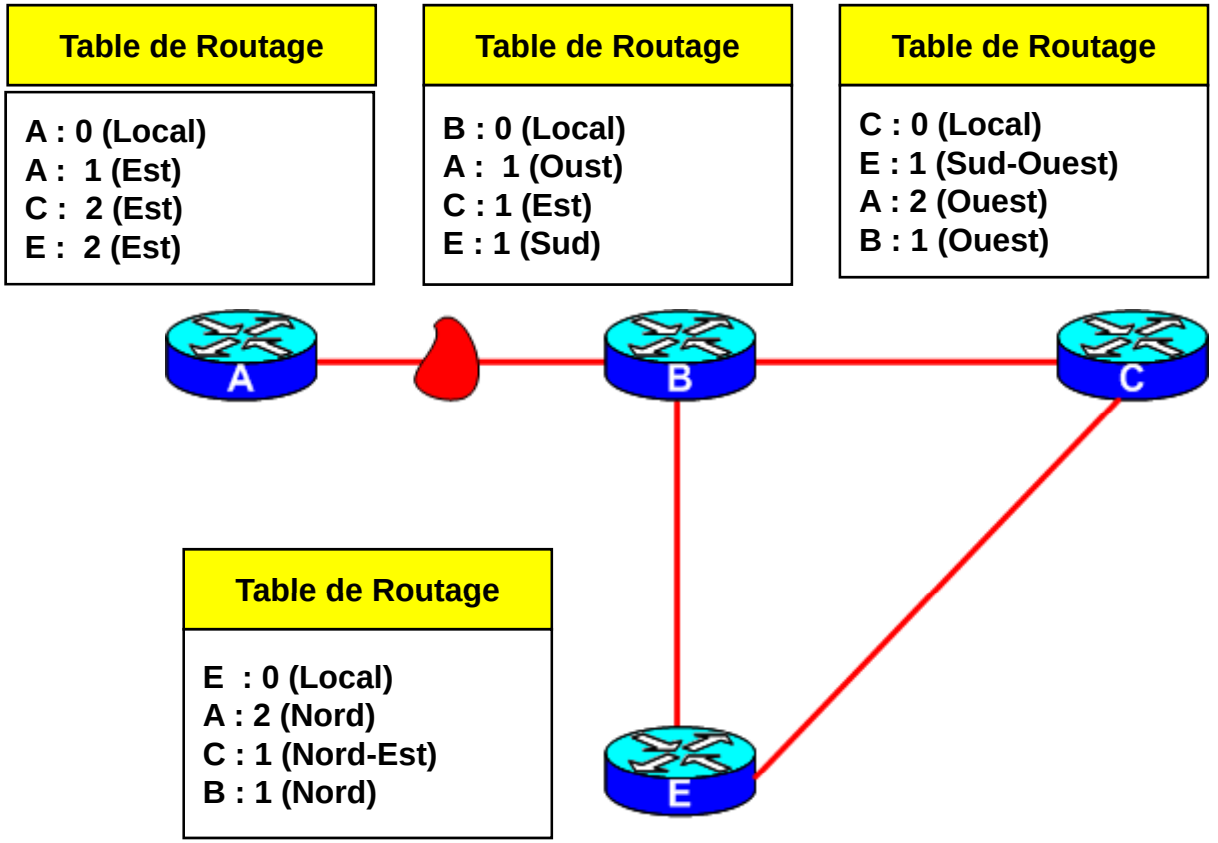
      → annoncer les destinations que l'on parvient à atteindre via d'autres lignes normalement

      → annoncer les destinations que l'on parvient à atteindre via cette ligne en indiquant une distance infinie
  - Avantage : permet d'annoncer plus rapidement les mauvaises nouvelles que split horizon simple

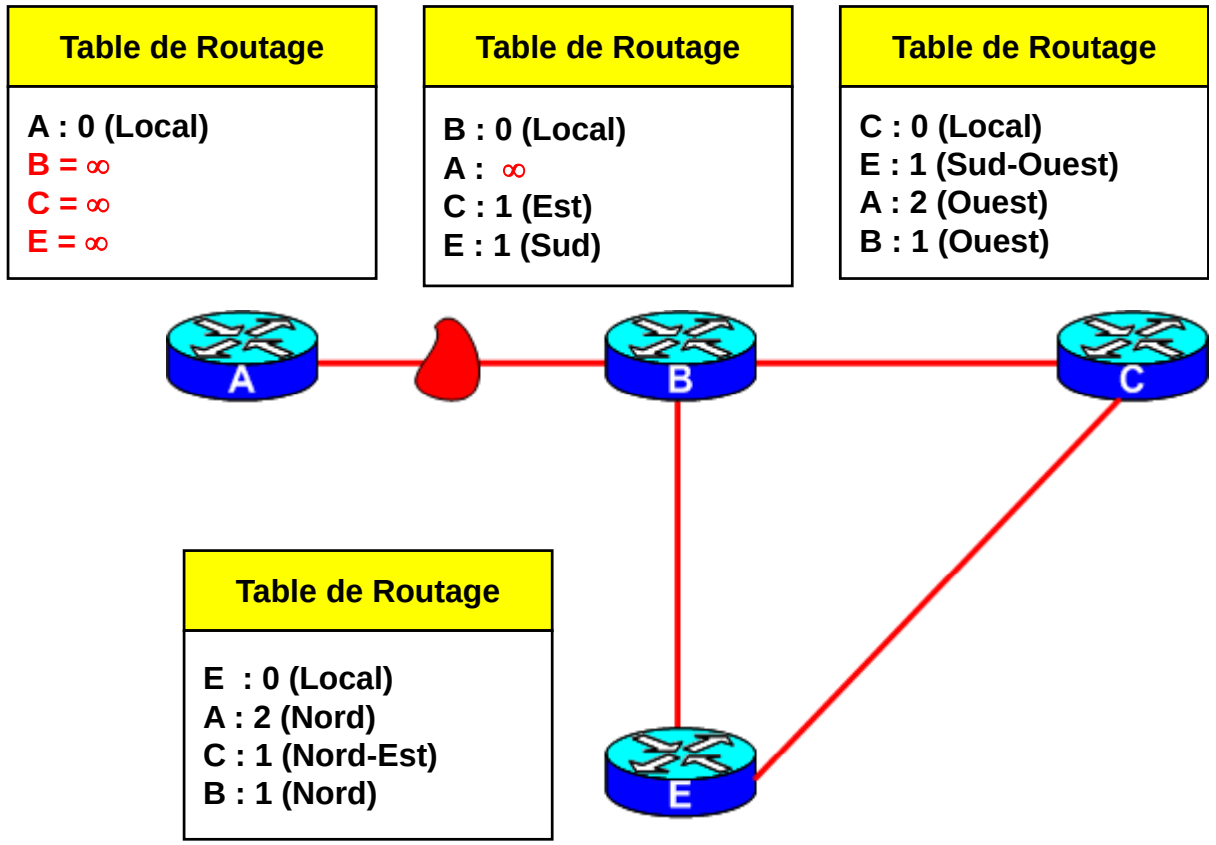
- Pb de comptage à l'infinie qui font intervenir plus de deux routeurs

  → On suppose l'exemple suivant dans lequel, un problème de coupure de la ligne est arrivée entre A et B.

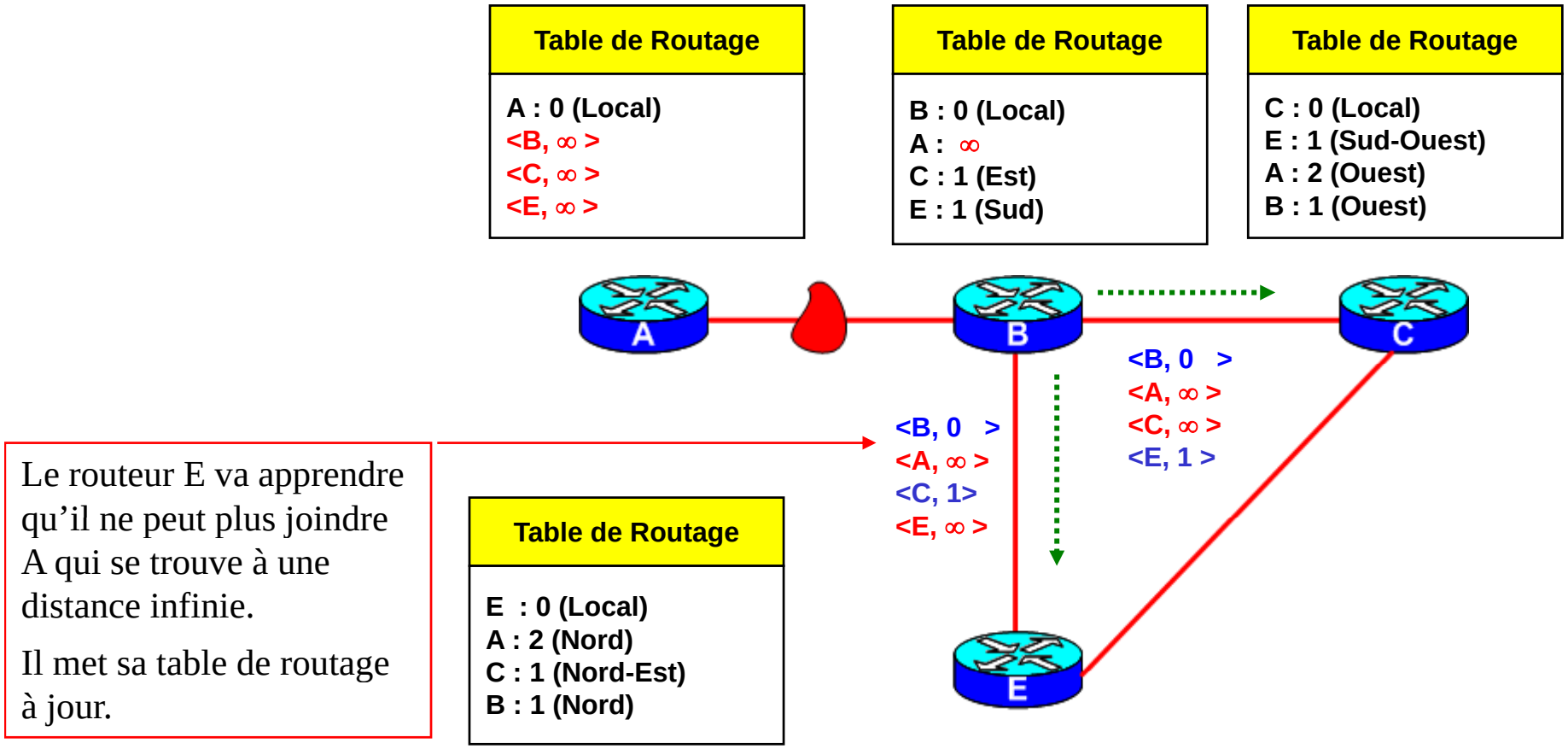
  → Les tables de routages sont celles avant que les routeurs A et B se sont rendus compte de la coupure du lien



- Les tables de routages après que les routeurs A et B se sont rendus compte de la coupure du lien



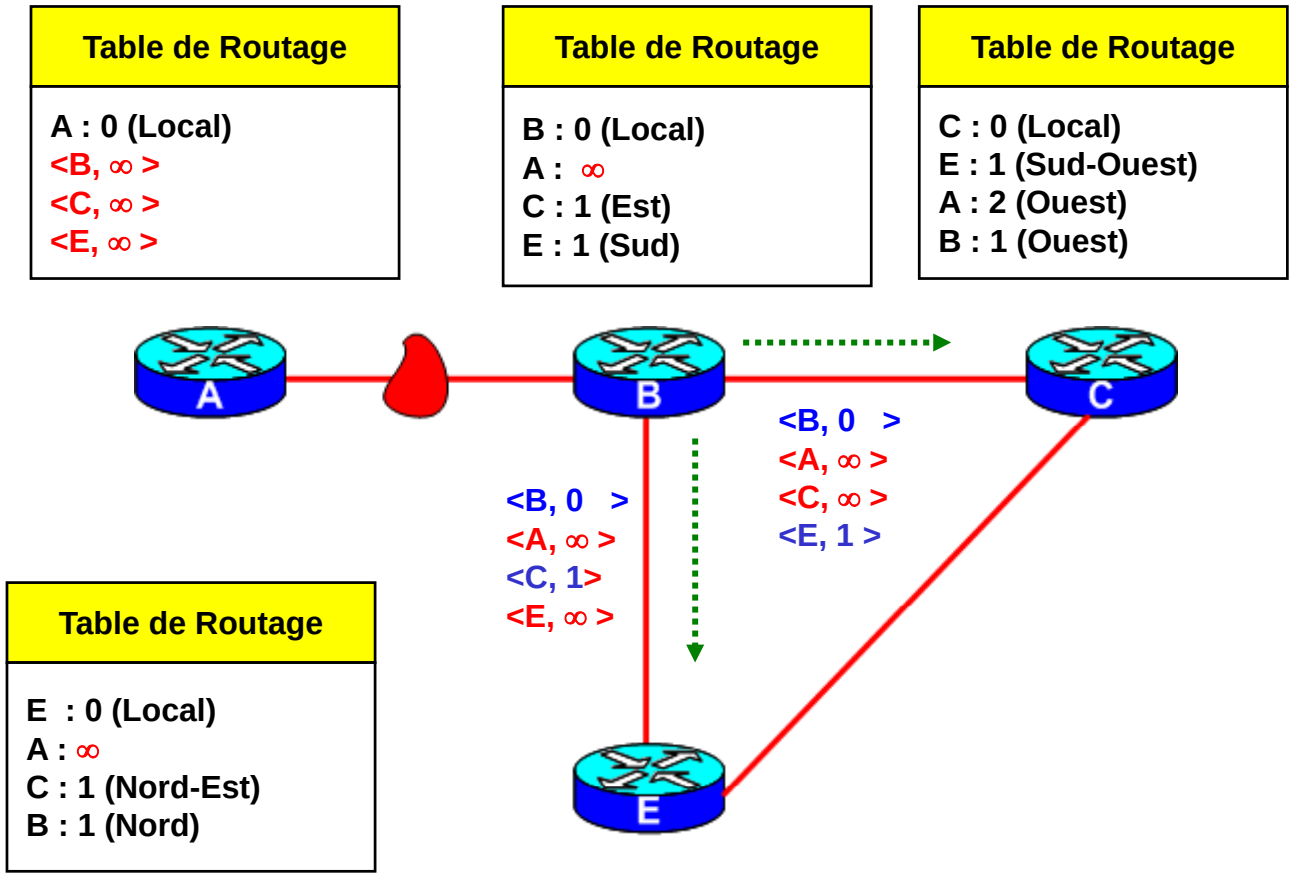
- Supposons qu'on travaille avec horizon partagé avec empoisonnement
- le premier qui se rend compte du problème est B **:**
  - B envoie ses vecteurs de distance



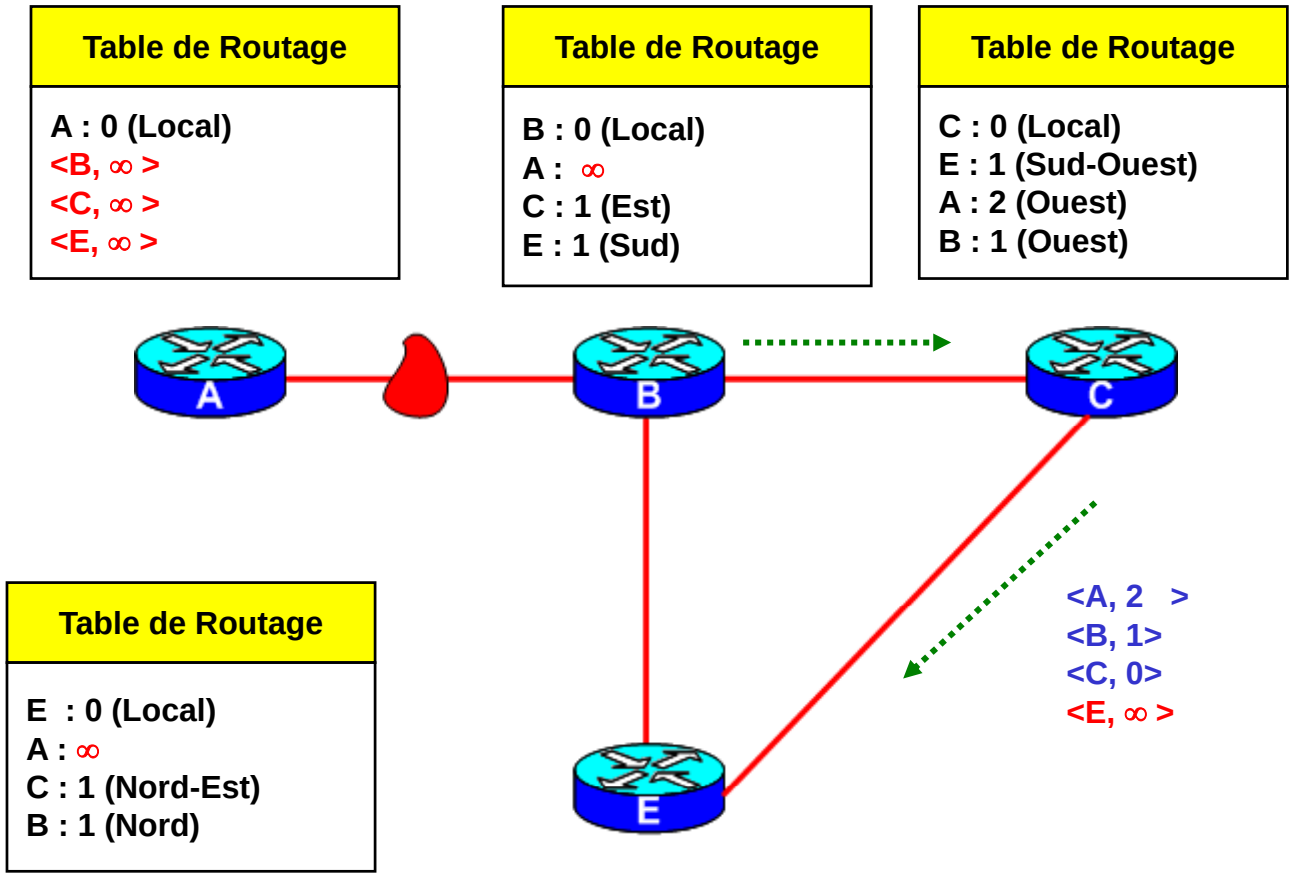
Le routeur E va apprendre qu'il ne peut plus joindre A qui se trouve à une distance infinie.

Il met sa table de routage à jour.

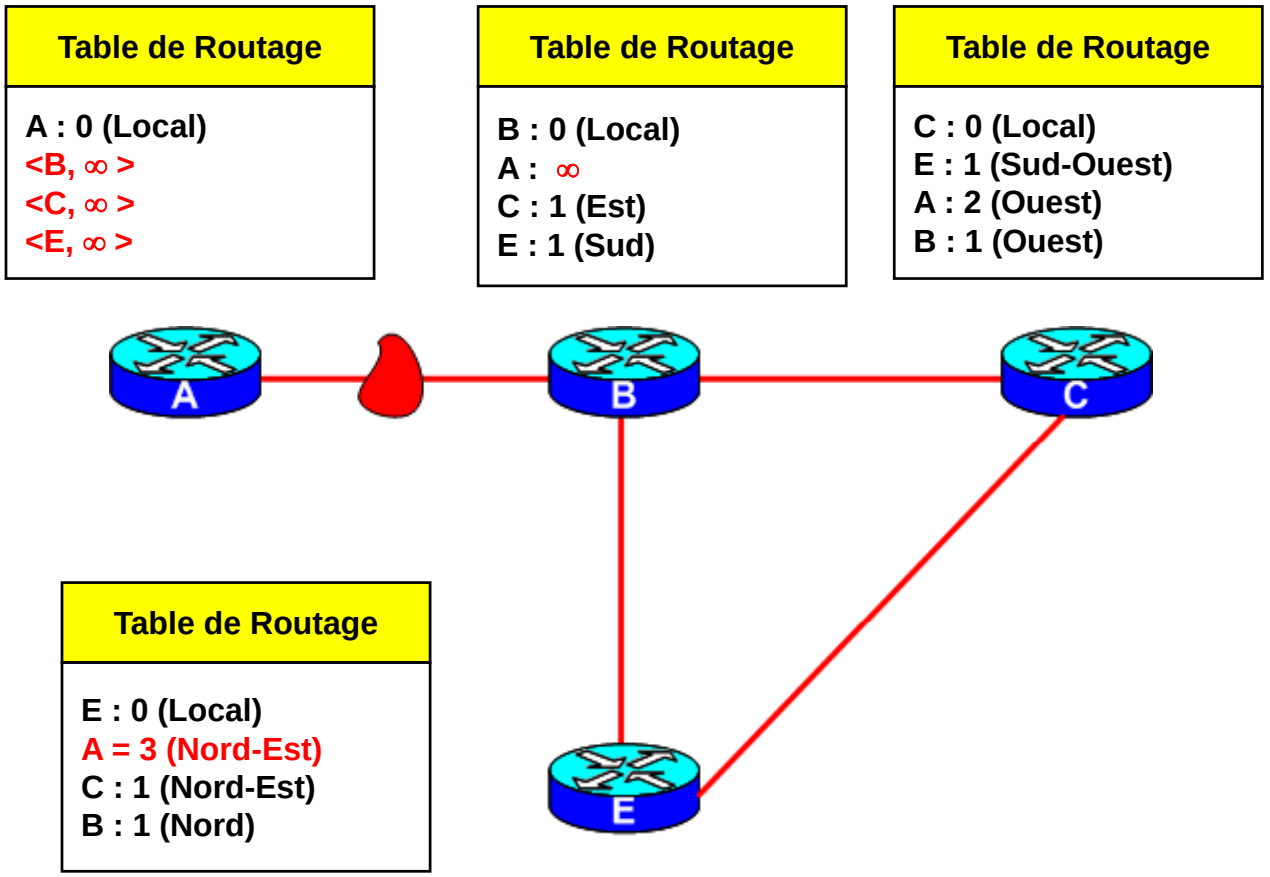
- le vecteur de B arrive en E mais pas encore à C



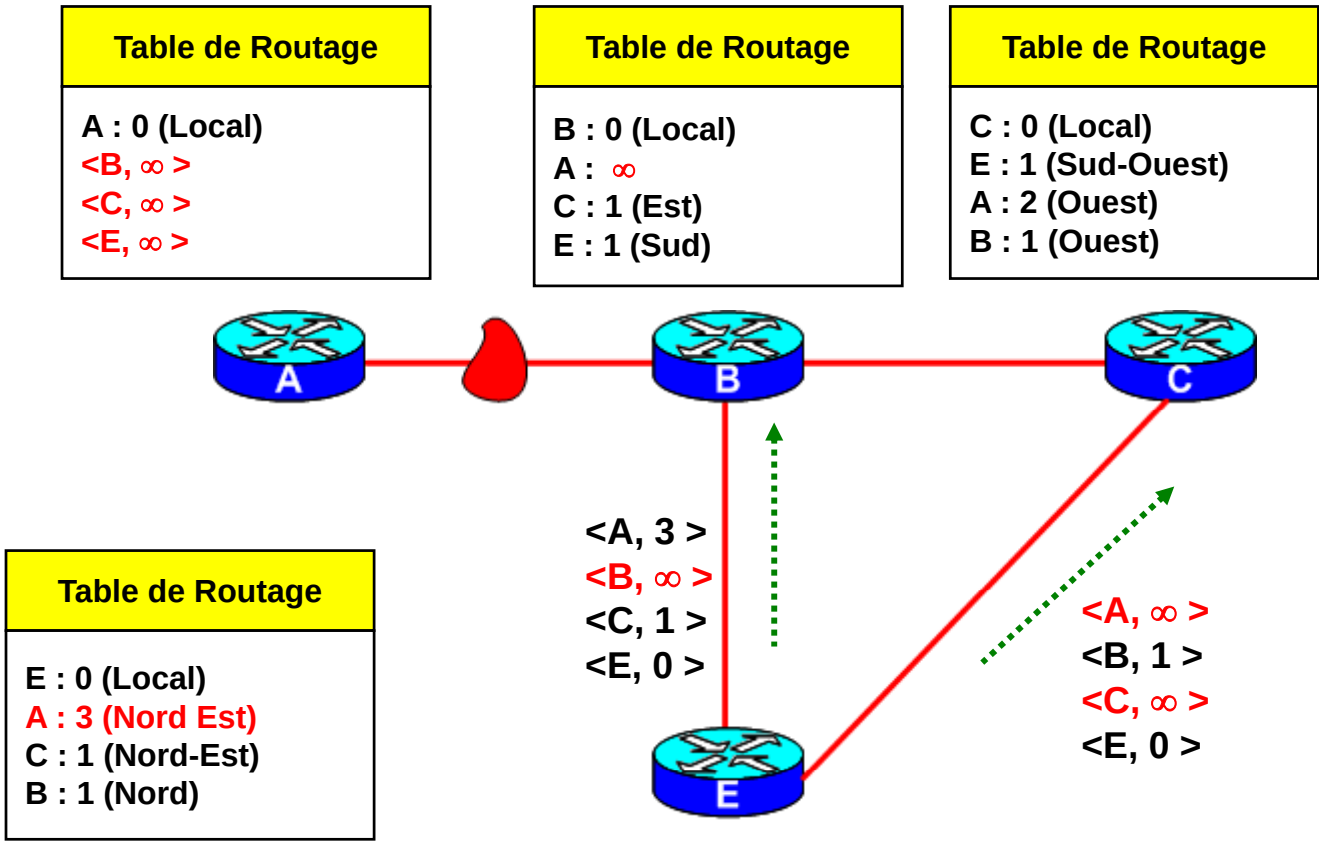
- *avant que le vecteur de B* n'arrive en C, C envoie son vecteur à E
  - **Malheureusement, toute suite après, le routeur E peut recevoir un vecteur de distance de C avant que C n'ait reçu un vecteur de distance de B**
  - Dans ce cas la, C va envoyer un vecteur de distance A=2; **B=** 1 **; C=0 ; E=** ∞ annonçant à **E qu'il peut joindre A avec une distance de 3**

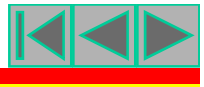


Après réception du vecteur de C par le routeur E

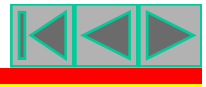


Un nouveau problème de comptage à l'infini





Routage avec état de liaisons

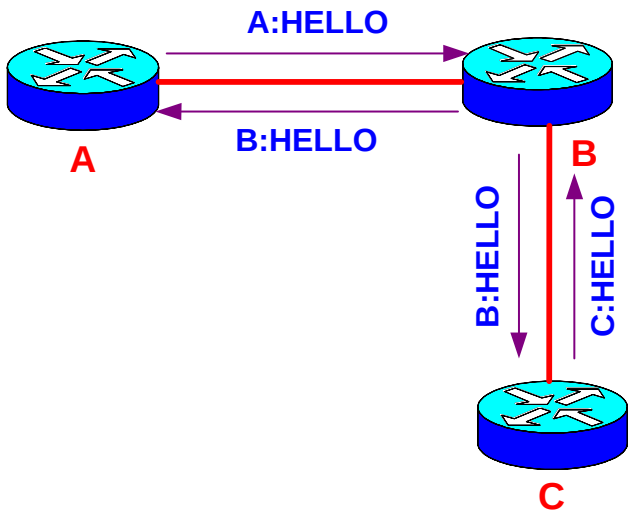


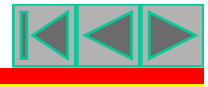
- Idée de base : Plutôt que d'envoyer uniquement la distance vers chaque destination, diffuser sur tous les routeurs une carte complète du réseau
  - ☞ *Problèmes à résoudre : Comment construire cette carte ?*
    1. Chaque routeur doit découvrir ses voisins
    2. Chaque routeur doit mesurer le délai de la ligne vers chacun de ces voisins
    3. Chaque routeur envoie un paquet indiquant ses proches voisins à tous les routeurs du réseau
    4. Les routeurs assemblent les paquets reçus et utilisent l'algorithme de Dijkstra pour calculer le chemin le plus court

❶ Premier problème à résoudre :

- Comment un routeur peut-il connaître ses voisins ?

- Solution : Echange de messages HELLO
  - Envoyer périodiquement un paquet spécial HELLO qui indique l'adresse du routeur sur chaque ligne
  - Les voisins répondent en envoyant leur adresse
  - Deuxième utilité de ce message :
    - L'envoi périodique permet de vérifier que la ligne reste active

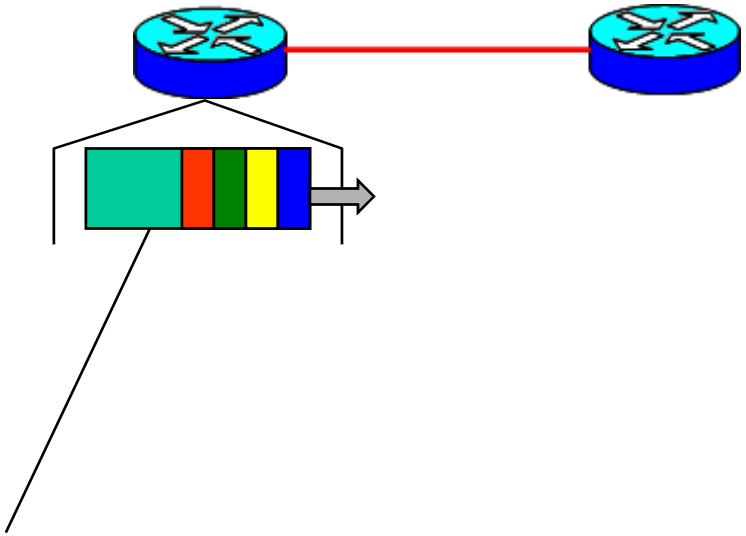




❷ Deuxième problème : Comment mesurer le délai des lignes ?

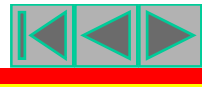
- Via le mécanisme HELLO
- Permet de mesurer le délai aller-retour
- En divisant par 2 on obtient le délai de la ligne
- Question intéressante : Le délai doit-il tenir compte de la charge actuelle de la ligne ?
  - SI « OUI » ALORS : Le routage pourra prendre la charge indirectement en compte, mais le routage risque d'être instable
  - SI NON : Le routage sera plus stable, mais certaines lignes risquent d'être mal utilisées

- Comment tenir compte de la charge de la ligne dans la mesure du délai ?



- Placement du paquet HELLO
- à la tête du buffer :
  - délai mesuré par HELLO sera indépendant de la charge de la ligne
- à la queue du buffer :
  - (comme si il s'agissait d'un paquet normal) :
  - Délai par HELLO mesuré sera fonction de la charge de la ligne

- Buffer de sortie du routeur contient des paquets en attente de transmission.
  - Si la charge est élevée : le buffer contiendra beaucoup de paquets en attente en moyenne
  - Si la charge est faible : le buffer contiendra peu de paquets en moyenne



- Comment déterminer la topologie du réseau ?

A. En découvrant ses voisins, chaque routeur découvre une petite partie de la topologie

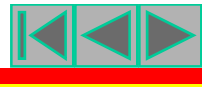
B. En assemblant ces petites parties, on peut construire la topologie complète

C. chaque routeur résume sa topologie locale en un Link State Packet (LSP) contenant :

   - ☞ Identification du routeur (adresse du routeur)
   - ☞ Couples (adresse du voisin, délai pour atteindre ce voisin)

D. Quand construire les LSPs et les diffuser ?

   1. Périodiquement (beaucoup plus faible par rapport au vecteur de distance)
   2. Ou en cas de problèmes (ligne devenant active ou en panne ou délai changeant fortement)



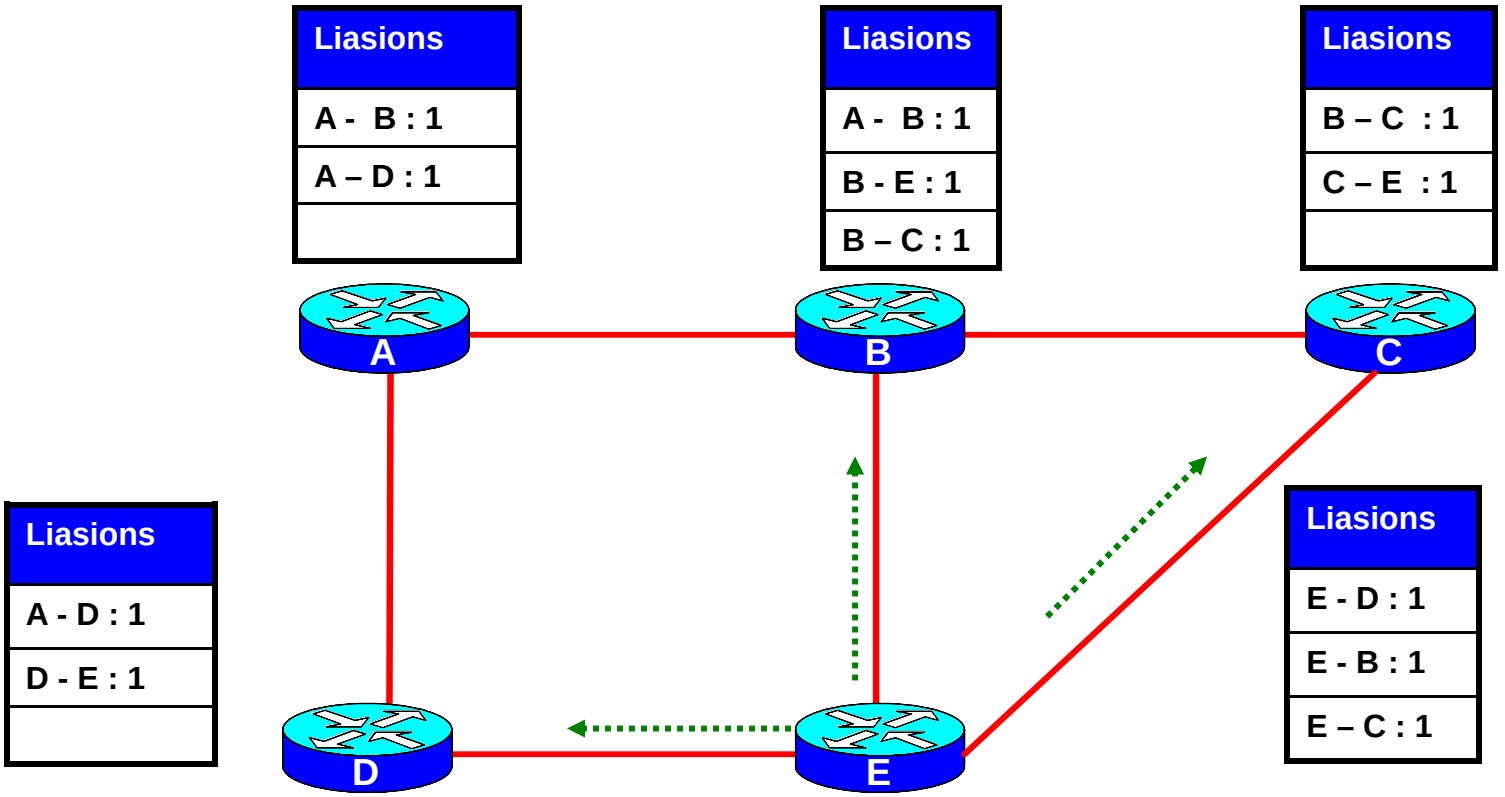
- Comment distribuer la topologie ? (sur quelle base s'appuyer pour envoyer les LSP)

❶ Première solution : En s'appuyant sur les tables de routage

- **Impossible** car ces tables n'existent pas ou risquent d'être incohérentes lorsque l'on commence à diffuser la topologie

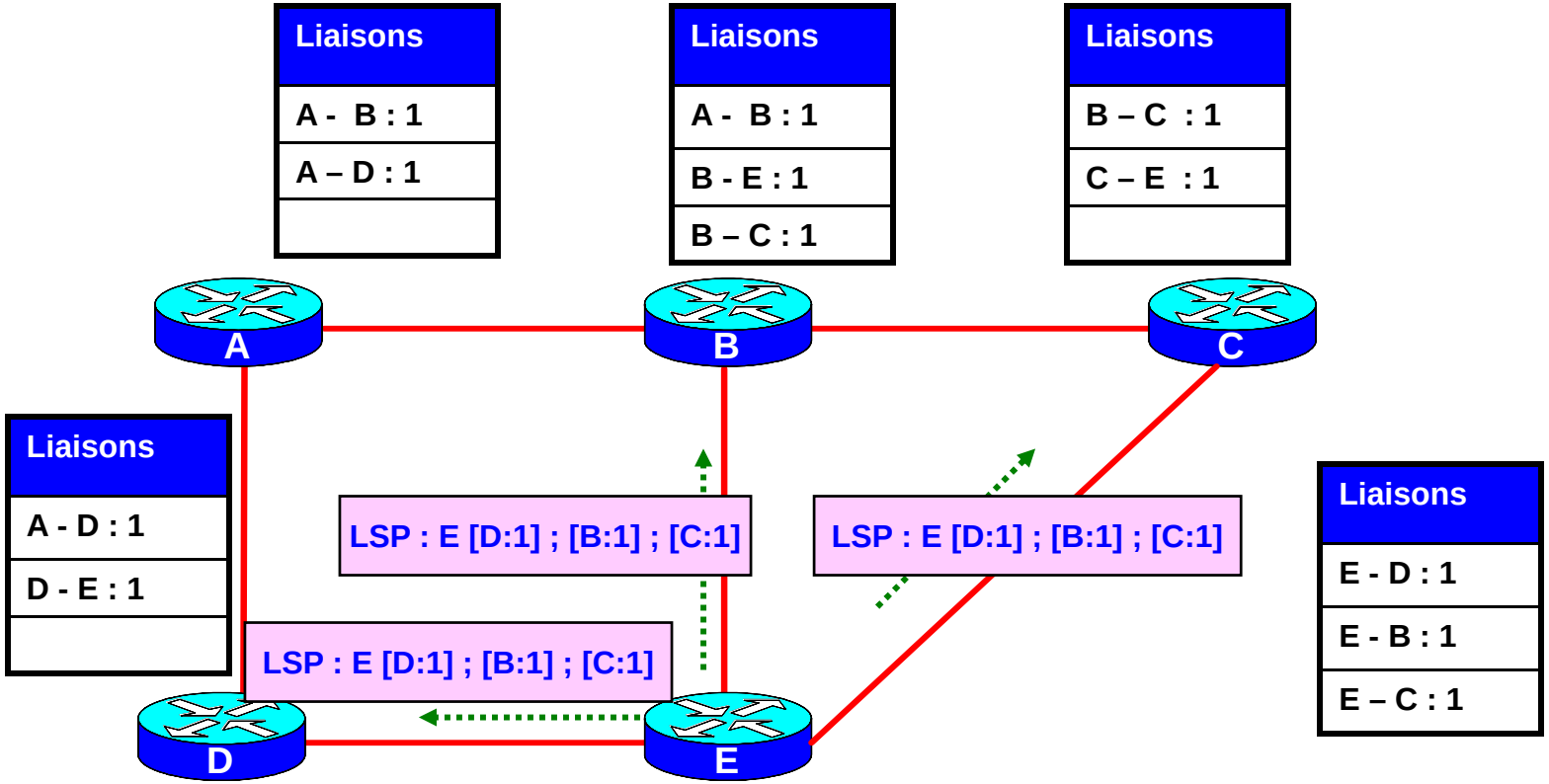
❷ Deuxième solution : Sans tenir compte de la table de routage

- **Principe :** Assurer que la topologie soit reçue par chaque routeur du réseau sur la base de la solution suivante **:**
  1. Chaque routeur place sa topologie locale dans un paquet spécial(LSP) et envoie ce paquet sur toutes ses lignes de sortie
  2. Tout routeur qui reçoit un LSP d'un autre routeur :
     - Stocke le contenu en mémoire
     - Et diffuse ce LSP sur ses lignes de sortie (sauf celle d'où le LSP provient évidemment)



1. Supposons que ici on va mesurer le nombre de sauts pour joindre un routeur pour que l'exemple soit facile
2. Au niveau du routeur A : Il va connaître seulement l'existence de la ligne A-B et de la ligne A-D avec des distances égales à l'unité (délai une unité)
3. Idem pour les routeurs C et D.
4. les routeurs B, E auront des tables à trois entrées

Diffusion des LSPs



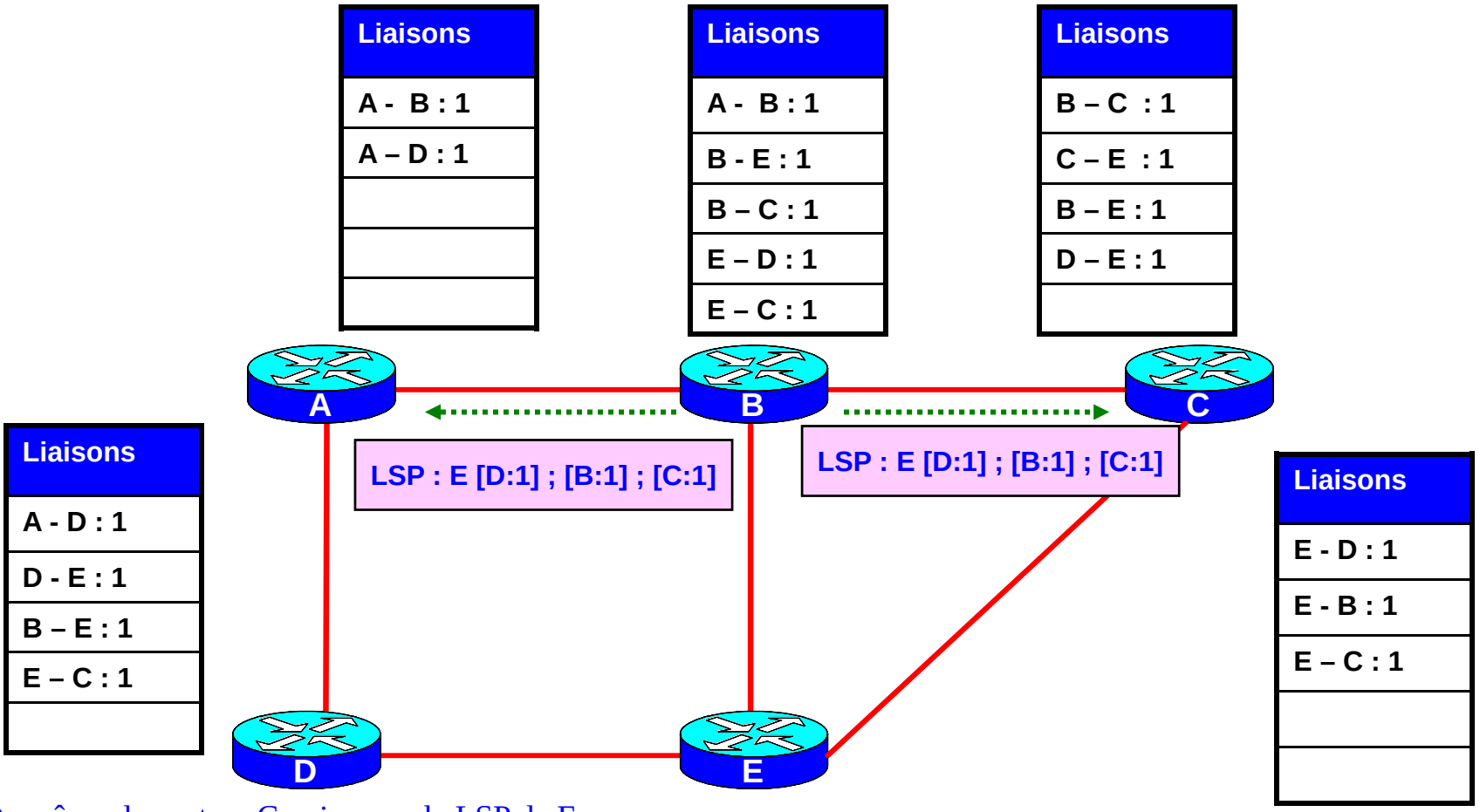
➢Supposons que **E** commence à diffuser le premier son LSP sur ses ports

- Contenu du LSP de E :
  - émetteur du LSP (⇔ E) et topologie locale du réseau autour de E : **E (adresse) , D:1; B:1 ; C:1**

Lorsque le LSP de E est reçu par **B**, il conserve ce LSP dans sa mémoire,

- il profite pour mettre sa table à jour.
- B sait maintenant qu'il existe un routeur D à un délai 1 de E et un routeur C à un délai 1 de E.
- En outre, B va permettre aux autres routeurs de réseau de connaître la topologie local de E, car il va diffuser le LSP de E sur ses sorties vers A et C.

Diffusion des LSPs –2-



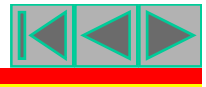
De même, le routeur C qui a reçu le LSP de E

- ✓ Va stocker ce LSP dans sa mémoire,
- ✓ Mettre sa table à jour
- ✓ Et diffuser ce LSP sur ses portes de sorties sauf celui d'où il est venu. Donc vers B.

Chaque routeur rediffuse le LSP reçu sur toutes ses lignes sauf celles d'où le LSP provient

- Comment éviter qu'un LSP ne boucle en permanence ?

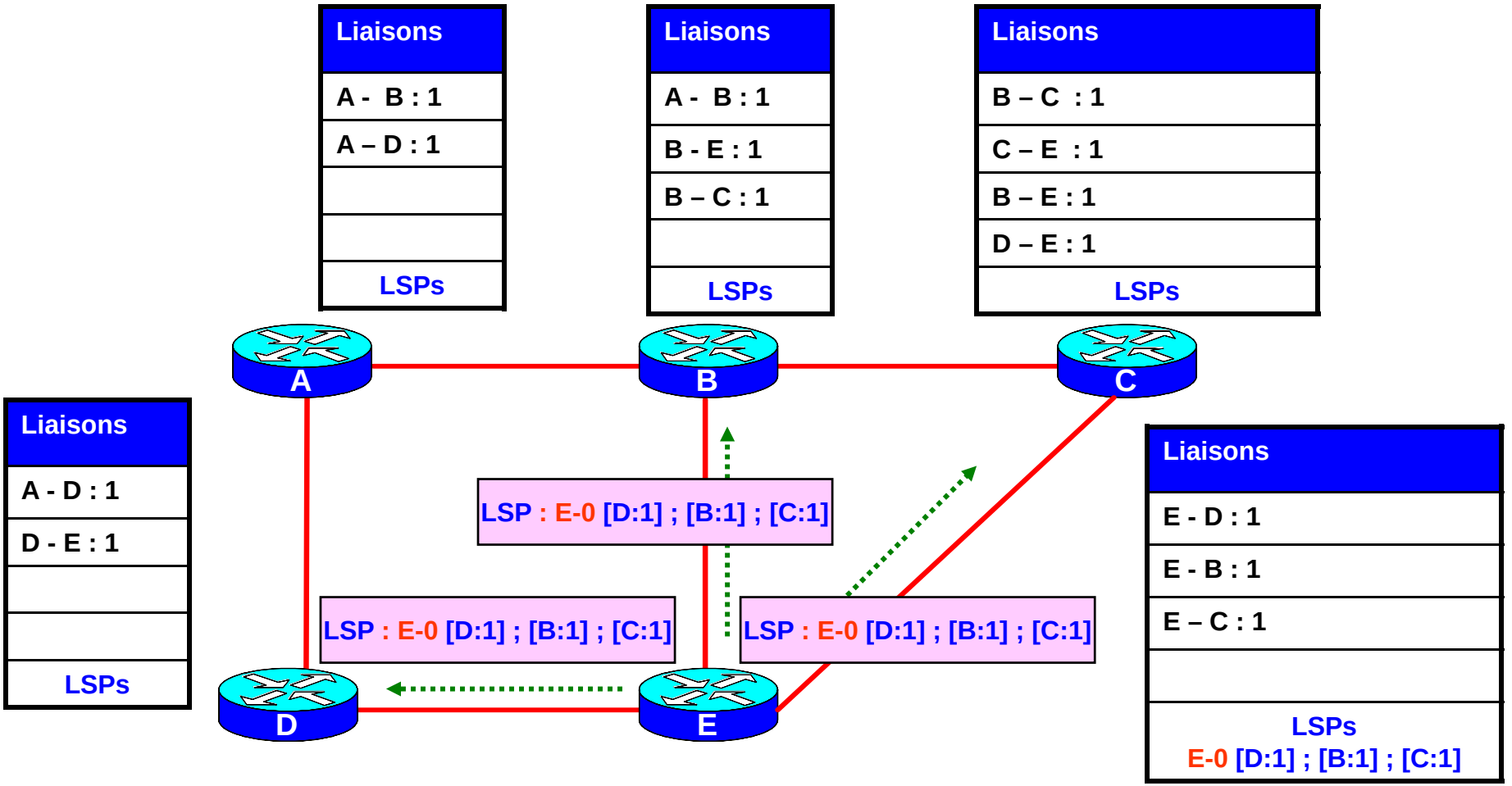
Diffusion des LSPs –3-



- Comment résoudre le problème du bouclage ?
  - Idée : Au niveau d'un routeur, il faut éviter de rediffuser un LSP qu'il a reçu précédemment sur une autre ligne.
  - Solution pour cela, un routeur doit conserver le dernier LSP émis par chaque routeur du réseau.
  - Principe :

    1. Chaque LSP contient :
       - → un numéro de séquence : incrémenté par l'émetteur à chaque nouveau LSP
       - → les couples « routeur:distance » pour tous les routeurs voisins du routeur qui émet le LSP
       - → une identification du routeur émetteur du LSP
    2. Chaque routeur stocke le LSP le plus récent reçu de chaque routeur du réseau (⇔ NumSéq. est strictement supérieur que celui qui est dans sa mémoire)
    3. Un LSP est traité et rediffusé uniquement s'il est plus récent que le LSP qui était dans la mémoire du routeur

Diffusion des LSPs –4-



Initialement la table des LSPs est vide

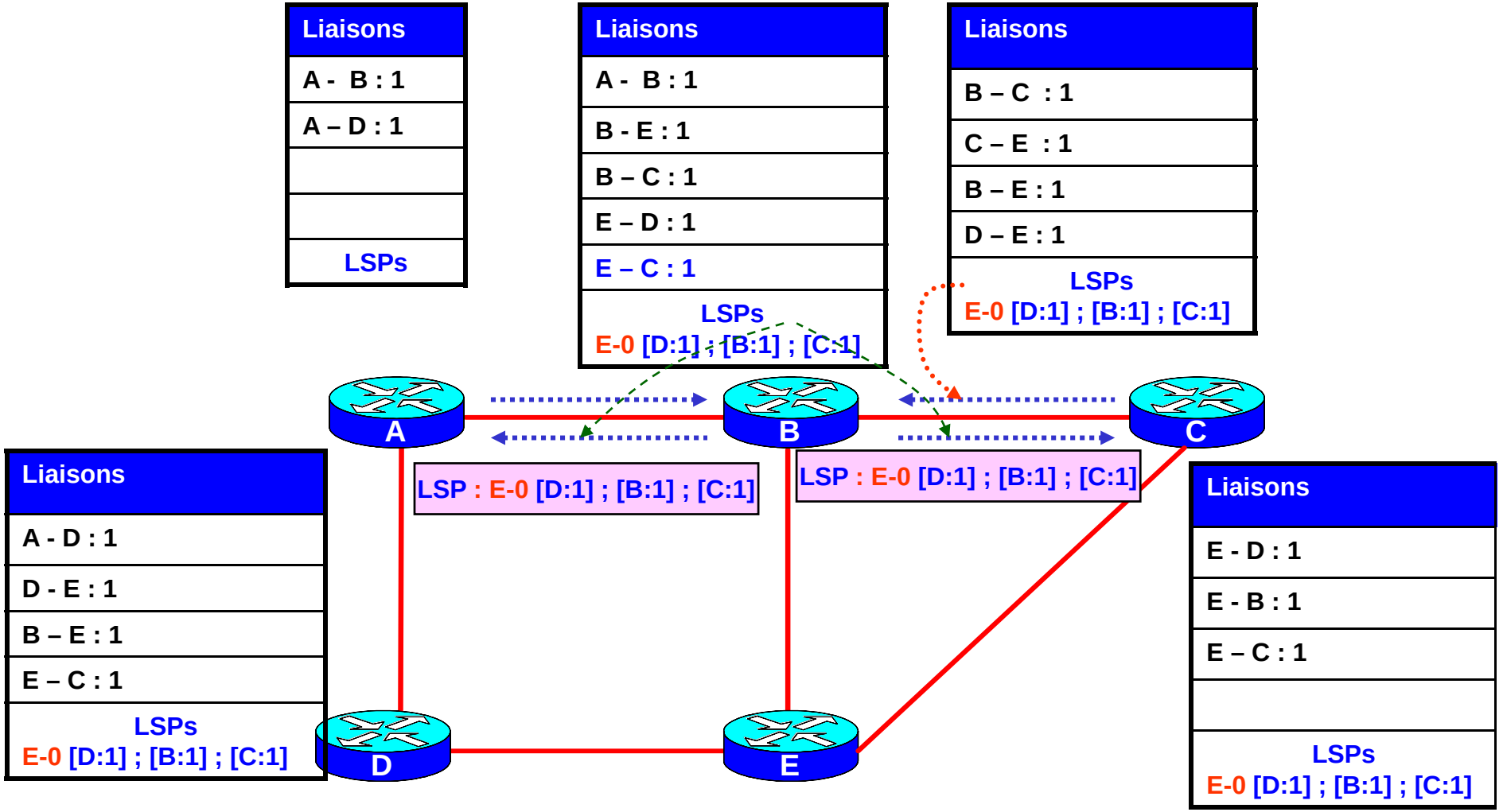
E : envoie son LSP sur tous ses ports

Le LSP envoyé par E est maintenant :

LSP : E – 0 (numéro de séquence) , [D:1] ; [B:1]; [C:1]

Vers B, vers D et vers C

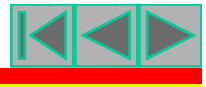
Diffusion des LSPs –5-



Grâce à la table des LSPs

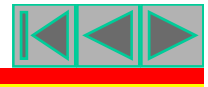
- A peut détecter qu'il reçoit le même LSP via D et via B
- B peut détecter qu'il reçoit le même LSP de C et de E
- C peut détecter qu'il reçoit le même LSP de E et via B

Diffusion des LSPs –6-



- Problème:
  - Un routeur se plante puis redémarre
  - Il envoie son LSP avec numéro de séquence = 0
  - Si un ancien LSP de ce routeur existait dans le réseau, les autres routeurs ne propageront pas le nouveau LSP

- Solution :
  - Ajouter un champ "age" dans les LSPs (fixé à une valeur de 10 par exemple) lorsque le routeur génère le LSP)
  - le champ age « est » décrémenté toutes les N secondes
    - à l'intérieur des tables de LSPs des routeurs
  - Un LSP avec « age = 0 » est considéré trop vieux et sera supprimé
    - Chaque routeur envoie périodiquement un nouveau LSP avec « age > 0 » pour garantir la présence du LSP dans le réseau
    -



- **Améliorations :**

1. Eviter qu'un LSP passe deux fois sur une ligne
   - lorsqu'un LSP est reçu, le routeur attend quelques secondes avant de le rediffuser
     - si le même LSP arrive d'une autre ligne, on a évité une diffusion
     - si après ce temps le LSP n'est pas arrivé, on le diffuse
2. Garantir la transmission correcte des LSPs
   - CRC dans chaque LSP pour détecter les erreurs de transmission
   - la réception de chaque LSP est acquittée par le routeur qui le reçoit sur la ligne où il l'a reçu
     - si l'acquit n'arrive pas, le LSP est réémis
3. *Un routeur qui démarre peut demander à un voisin de recevoir les LSPs stockés par ce voisin*

- ***Exemple :***

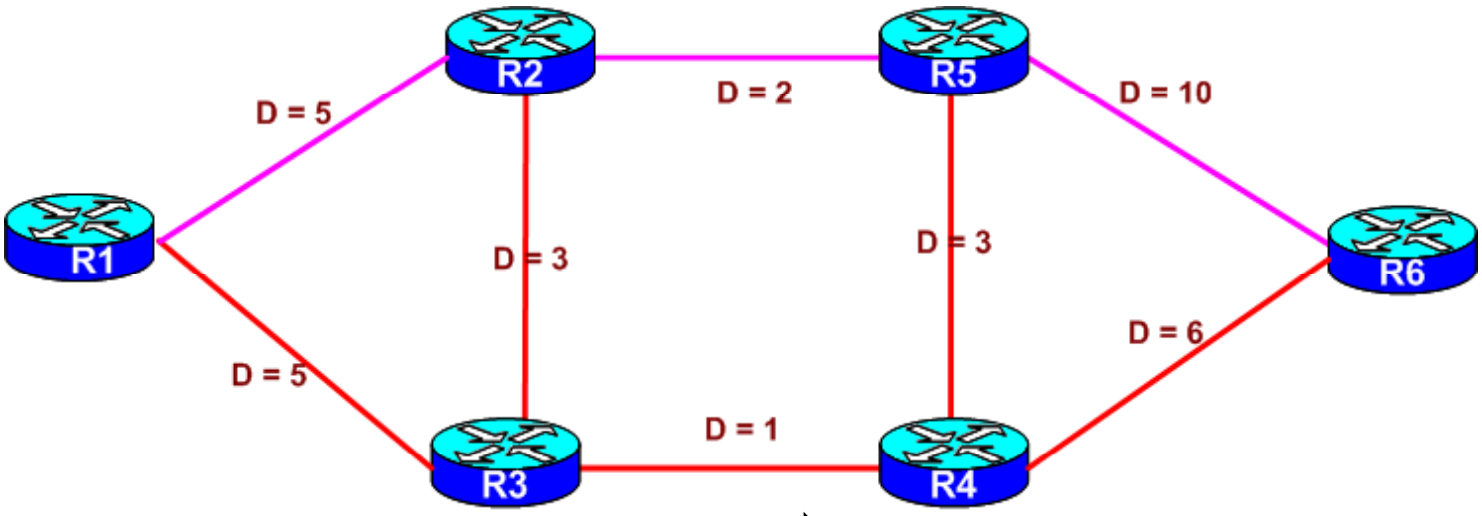


Table de routage
R1 : Ouest
R2 : Nord
R4 : Est
R5 : Est
R6 : Est

Liaisons
R3-R1 : 5
R3-R2 : 3
R3-R4 : 1
R4-R5 : 3
R4-R6 : 6
R2-R5 : 2
R2-R1 : 5
R5-R6 : 10

Algorithme de Dijkstra –2-

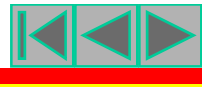


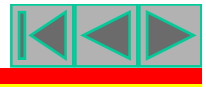
Table de routage
R1 : Ouest
R2 : Nord
R4 : Est
R5 : Est
R6 : Est

Liaisons
R3-R1 : 5
R3-R2 : 3
R3-R4 : 1
R4-R5 : 3
R4-R6 : 6
R2-R5 : 2
R2-R1 : 5
R5-R6 : 10

Arbre de R3 ? Routeurs : [R1, R2, R4, R5,R6] ;

1. Sélection des Candidats; ⌦ Candidats : [R1(5) ; R2(3) ; R4 (1)]
   - Candidat choisi : R4 :
   - Nouvel Arbre : R3 → R4
2. Nouveaux Candidats ? : [R1(5) ; R2 (3) ; R5(R4 - 4) ; R6(R4 - 7)]
   - Candidat choisi : R2 ;
   - Nouvel Arbre : R3 → R4 et R3 → R2
3. Nouveaux Candidats ? [R1(5) ; R5(R4 - 4) ; **R5(R2 – 5),** R6 (R4 - 7),R1(R2-8)]
   - Candidat choisi : R5 ; on ajoute R4 → R5 :
   - Nouvel Arbre : R3 → R4 , R3 → R2 et R3 → R4 → R5
4. Nouveaux candidats ? [R1(5) ; R6(R4-7), **R6(R5 – 15**) , R1(R2-8)]
   - Candidat choisi : R1 : on ajoute : R3 -> R1
   - Nouvel Arbre : R3 -> R1, R3 → R4 , R3 → R2 et R3 → R4 → R5
5. Nouveaux candidats ? [R6(R4-7), **R6(R5 – 15**)]
   - Candidat choisi : R6 via R4: on ajoute : R3 -> R4 -> R6,
6. Arbre du R3 :

   R3 -> R4 -> R6, R3 -> R1, R3 → R4 , R3 → R2 et R3 → R4 → R5



Construction de l'arbre de recouvrement minimum

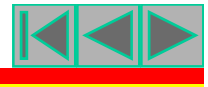
1. Initialement, l'arbre comprend uniquement la racine
2. Les routeurs adjacents sont placés avec les coûts de leur ligne dans une liste de candidates
3. Le routeur candidat avec le coût le plus faible est ajouté à l'arbre
4. On examine les voisins du routeur candidat choisi et on modifiera la liste des candidats si
   - un des voisins ne se trouvait pas dans la liste des candidats
   - un des voisins est un routeur de la liste des candidats, mais avec un chemin plus court que celui qui est dans la liste actuelle
5. L'algorithme se poursuit avec la nouvelle liste des candidats et s'arrête avec tous les routeurs dans l'arbre

Résumé : comparaison entre Vecteurs de distances et Link State Pakets

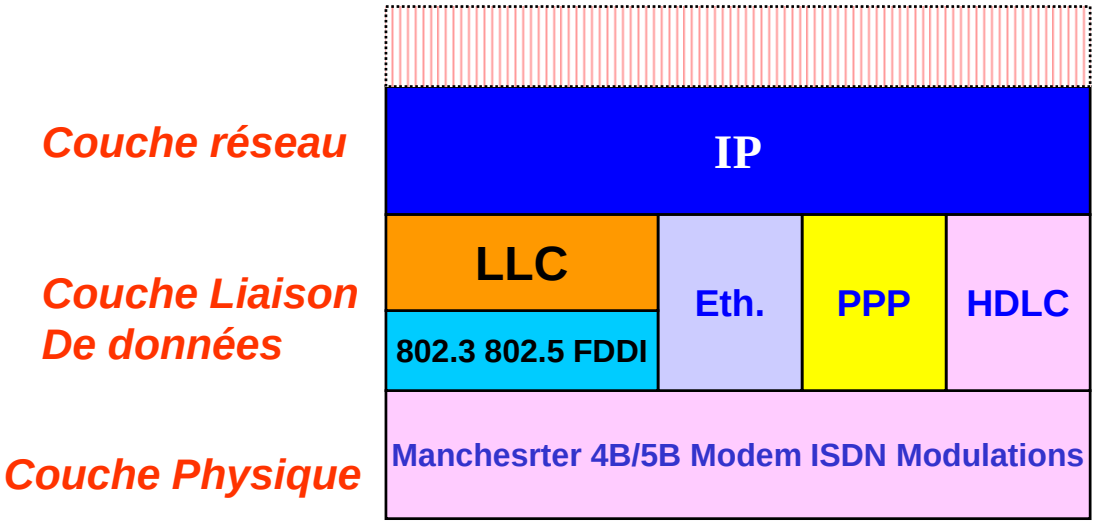
Algo. : Etat Liaison "Link State" « Dijkstra »	Vecteur de distance « Bellman – Ford »
• La topologie du réseau est distribuée sur tous les routeurs	• Un routeur n'a pas une connaissance complète de la topologie du réseau
• Le meilleur chemin (bout à bout) est calculé localement au niveau de chaque routeur	• Un routeur choisit le meilleur "next-hops" prochain saut pour chaque réseau de destination
• Le meilleur chemin end-to-end paths détermine les prochains sauts (next hops)	• Le meilleur chemin end-to-end résulte de la composition de tous les next-hop choisis
• Marche seulement si politique de routage est partagée et uniforme	• Chaque routeur peut appliquer sa propre politique de routage
• Exemples: OSPF, IS-IS	• Exemples: RIP, BGP
	• Une erreur dans la table se propage dans tous le réseau

Un routeur utilise la table des autres routeurs : donc, l'erreur se propage dans le réseau

Routage IP



- **Le protocole IP est le protocole réseaux le plus utilisé aujourd'hui.**
  - **C'est un protocole qui fonctionne en mode datagramme**
- **Il fournit un service sans connexion non fiable en s'appuyant sur quasiment tous les protocoles de liaison de données imaginable et possibles.**
- **Le seul effort que IP fait est d'acheminer les informations vers une destination,**
  - **mais il peut acheminer avec des pertes, sans pertes, avec des erreurs ou avec une modification de la séquence.**

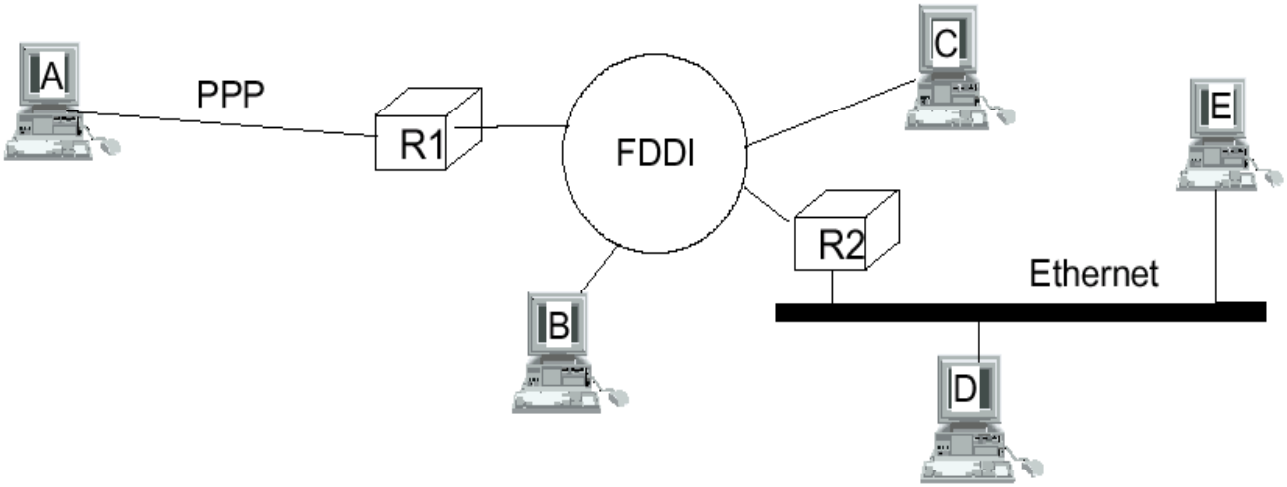


Couche réseau de l'Internet
fournit un service sans connexion non fiable

1. **perte possible de paquets**
2. **erreurs de transmission possibles dans les paquets**
3. **déséquencement possible des paquets**

- Fonctionnement en mode datagramme

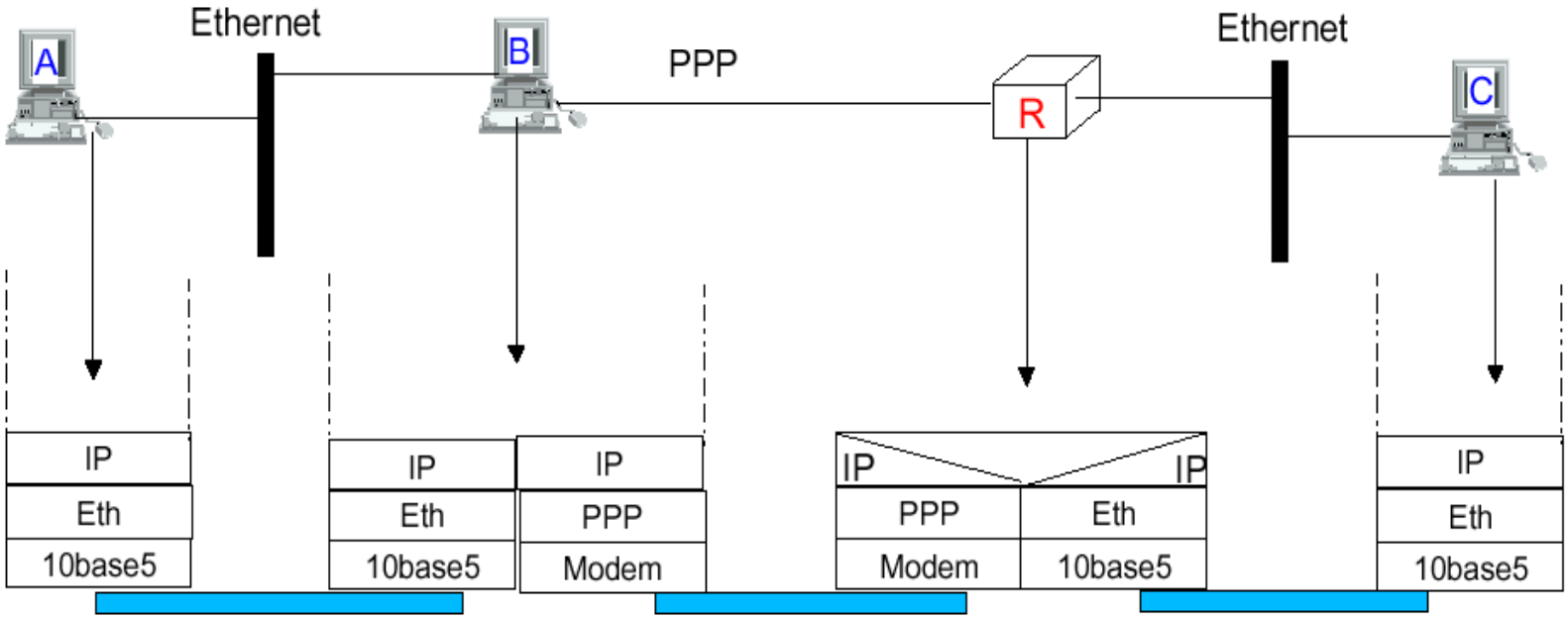
  ⇔ Les stations interconnectés vont s'échanger des paquets dans lesquelles, on trouve une adresse source et une adresse destination



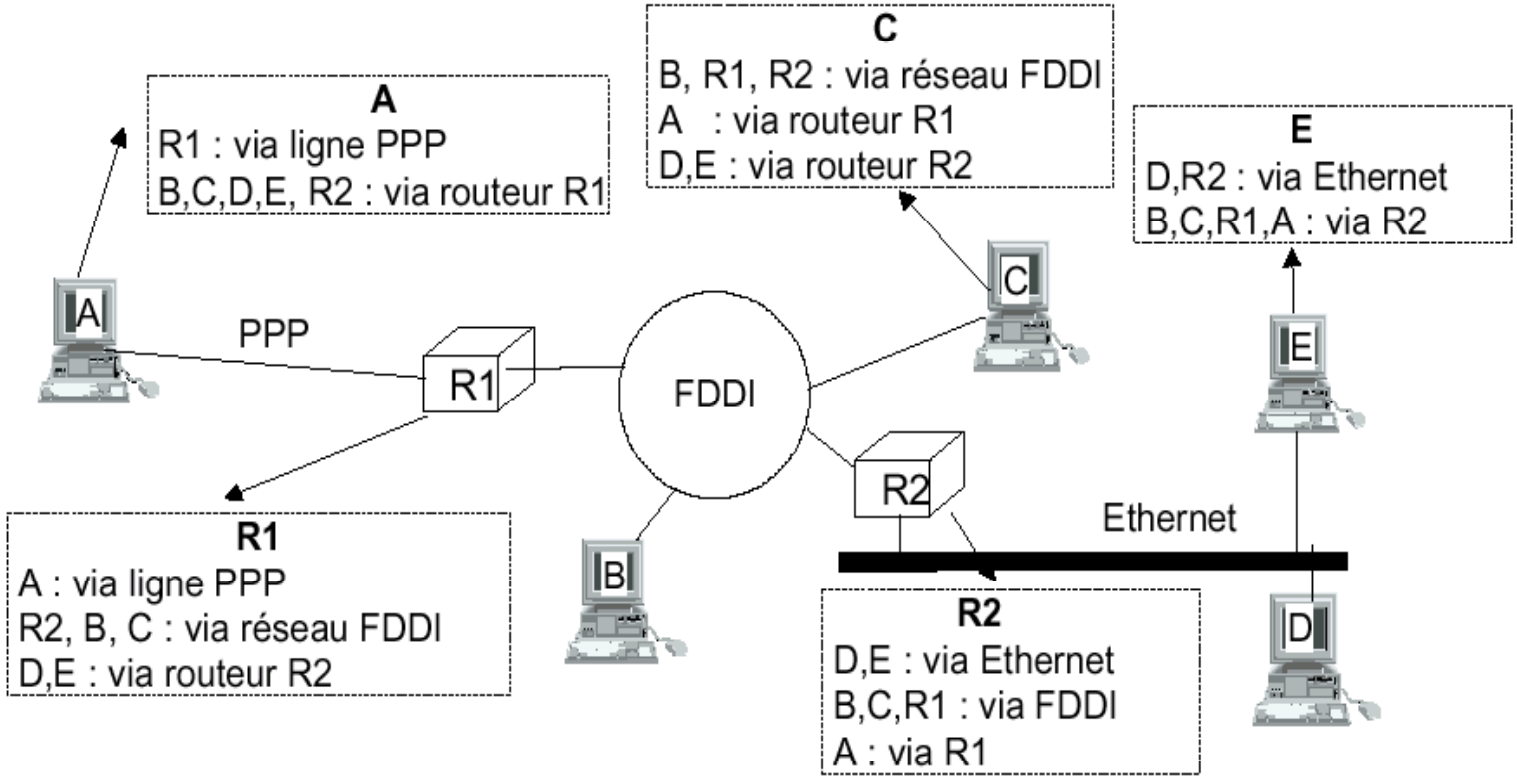
- Chaque station connectée au réseau est identifiée au moins par une adresse IP

  → une adresse IP (adresse IP = nombre de 32bits)

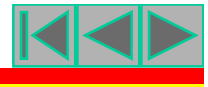
- Pour acheminer les informations, chaque station et chaque routeur doit savoir comment joindre les autres stations du réseau



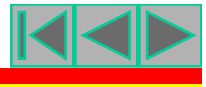
- Station terminale (machines A, B et C dans l'exemple): équipement capable de recevoir et d'envoyer des paquets IP , Mais uniquement recevoir des paquets IP qui leurs sont destinés
- Routeur : équipement capable de recevoir, envoyer et retransmettre vers leur destination des paquets IP (Et en général, un routeur dispose de plusieurs interfaces pour échanger des informations entre plusieurs réseaux)



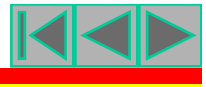
- Chaque station/routeur doit connaître :
  1. adresses joignables via la couche liaison de données
     ⇔ stations sur le même réseau local
  2. routeur intermédiaire permettant de joindre les stations et routeurs non accessibles directement



- Remarque :
  - Le routeur servira à passer d'un réseau physique vers un autre réseau physique différent;
  - ça peut être un routeur entre un réseau FDDI et un autre Ethernet
  - ou un routeur entre plusieurs réseaux Ethernet
    - ça permet de faire une séparation de trafic au niveau de la couche IP entre réseaux ou faire de la sécurité pour éviter que le trafic broadcast d'un réseau Ethernet passe sur une autre réseau Ethernet



- Rôle de l'adresse IP
  - identifier une station/routeur supportant IP
  - en pratique, une adresse IP identifie une interface sur une station ou un routeur
    1. l'interface est le point d'accès de la station/routeur à la couche liaison de données
    2. en général une station possède une interface
    3. en général un routeur possède plusieurs interfaces
- Comment attribuer les adresses IP ?
  1. Par construction, comme dans Ethernet
     - inutilisable en pratique, chaque routeur devrait connaître dans sa table de routage l'ensemble des adresses IP
  2. De façon hiérarchique :
     - Attribuer des blocs à des zones de réseaux, ceci permettra de résumer dans les tables de routage en une seule ligne
     - On ne demandera pas aux routeurs de connaître toutes les stations joignables, mais seulement les zones joignables



- Principes d'attribution des adresses IP
  - une adresse IP par interface sur chaque machine
  - une adresse IP comprend deux parties :
    1. l'identificateur du sous réseau (FDDI, TR, PPP, Ethernet auquel on est connecté)

       M bits de poids fort de l'adresse IP
    2. l'identificateur de la machine à l'intérieur du sous-réseau

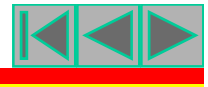
       32-M bits de poids faible de l'adresse IP

Exemple 10001010 00110000 00011010 00000001

id sous-réseau *id machine*

Notation :138.48.26.1/23 ou 138.48.26.1 255.255.254.0 (masque)

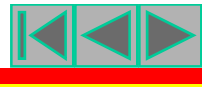
- toutes les stations faisant partie d'un sous-réseau peuvent directement s'échanger des trames par l'intermédiaire de la couche liaison de données



- Comment allouer les adresses IP ? (historiquement deux solutions)

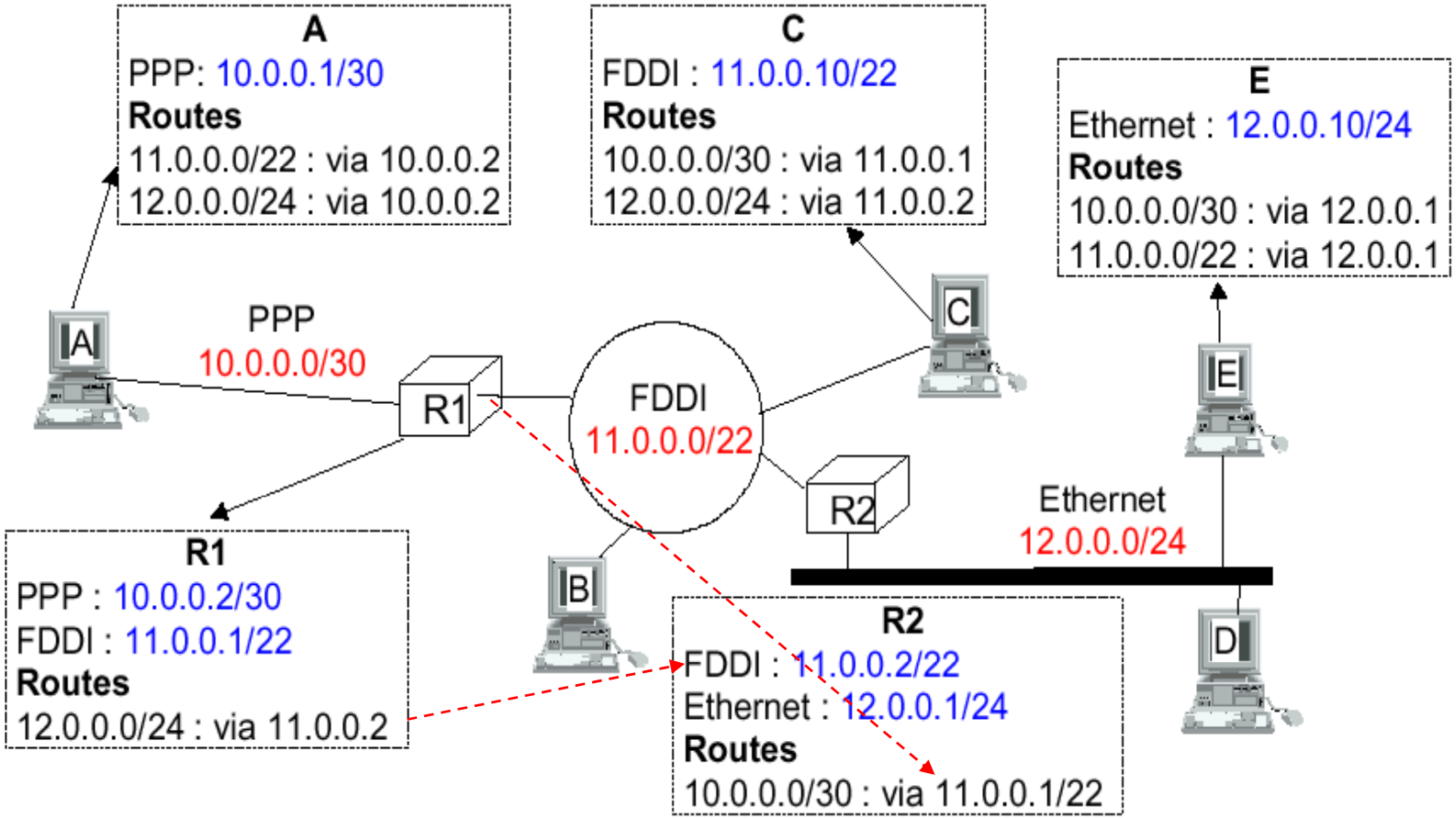
1. Première solution
   - Objectif : Assurer l'unicité des adresses IP
   - Règle
     - Toute organisation peut obtenir un sous-réseau IP unique pour ses ordinateurs parmi les adresses non encore allouées *(en s'adressant à un organisme aux E.U)*
   - En pratique, trois types de sous-réseaux
     - Classe A : sous-réseau avec masque de 8 bits
     - Classe B : sous-réseau avec masque de 16 bits
     - Classe C : sous-réseau avec masque de 24 bits

- Inconvénients
  - taille des différentes classes d'adresses trop rigide
    - -> gaspillage d'adresses
  - deux entreprises se connectant à l'Internet par le même fournisseur peuvent avoir des adresses complètement différentes
    - ->agrégation impossible

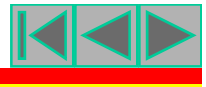


2. Deuxième solution

- Objectif :
  - assurer l'unicité des adresses en permettant une meilleure agrégation des routes
- Règles :
  - Seuls les fournisseurs d'accès Internet (ISP) peuvent recevoir des blocs d'adresses IP
    - la taille du bloc attribué dépend du nombre d'utilisateurs
  - Une organisation qui veut se connecter à Internet doit obtenir ses adresses IP de son fournisseur d'accès
- Avantage :
  - meilleure agrégation des adresses
- Inconvénient :
  - une organisation qui change de fournisseur d'accès devra changer les adresses IP de ses machines



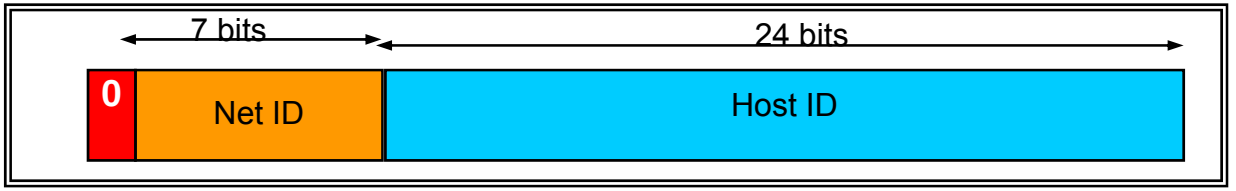
- Inconvénient de l'utilisation des sous-réseaux
  - gaspillage d'adresses : le sous-réseau contient souvient plus d'adresses que de machines physiquement connectées au réseau



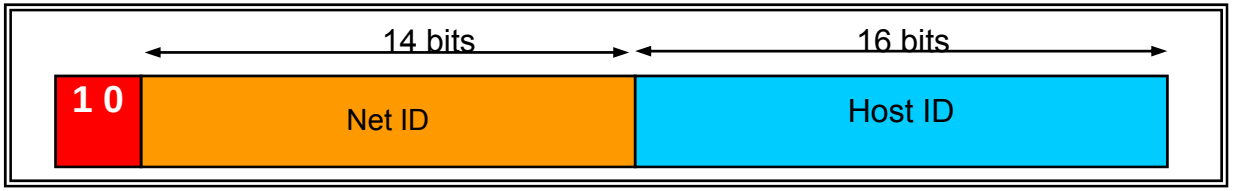
Annexe : pour mémo

- Il existe 5 classes d'adresses IP :

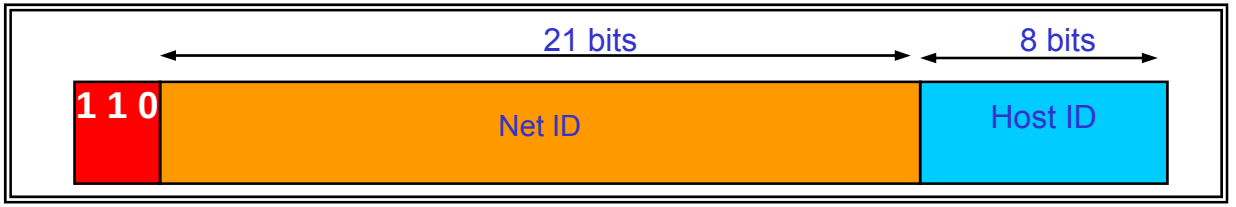
Classe A :



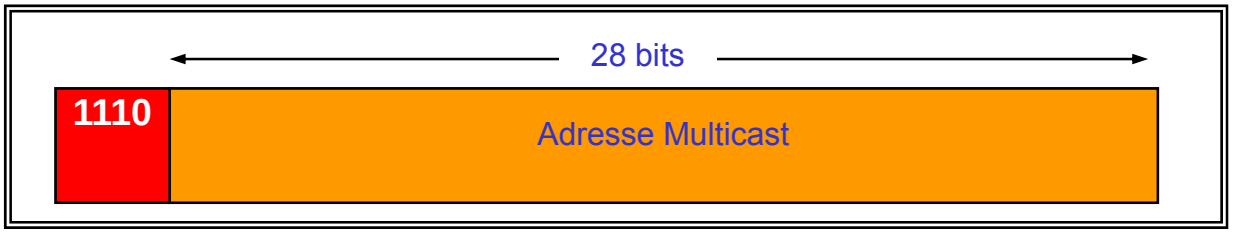
Classe B :



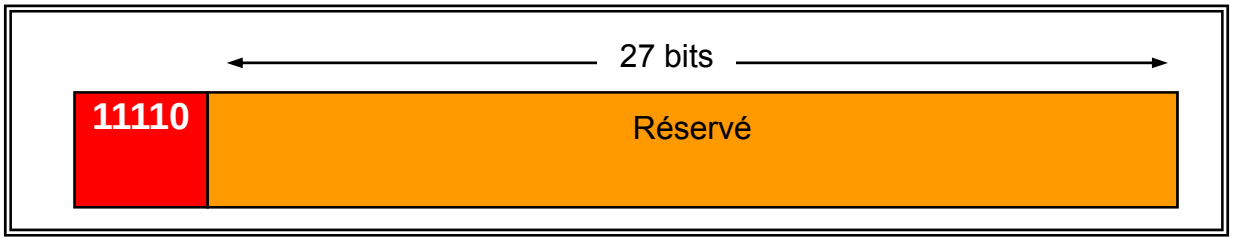
Classe C :

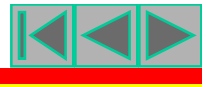


Classe D



Classe E





- Selon la valeur des bits du premier octet représentant l'adresse réseau IP, il est facile de déterminer la classe utilisée

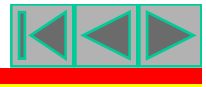
Classe	Gamme en notation décimale	Premier octet en binaire	Nb de réseaux	NB de noeuds
A	0.0.0.0 à **127**.255.255.255	**0** 0000000 et **0** 1111111	**126**	**16 777 214**
B	**128**.0.0.0 à **191**.255.255.255	**10** 000000 et **10** 111111	**16383**	**65534**
C	**192**.0.0.0 à **223**.255.255.255	**110** 00000 et **110** 11111	**2 097 151**	**254**
D	**224**.0.0.0 à **239**.255.255.255	**1110** 0000 et **1110** 1111		
E	**240**.0.0.0 à **247**.255.255.255	**11110** 000 et **11110** 111		

Gammes d'adresses IP en fonction des classes

- Adresses Privées :
  - Pour les réseaux non connectés à l'Internet, les administrateurs décident de la classe et de l'adresse NetID.
  - Cependant pour des évolutions possibles, il est fortement recommandé de servir des adresses non utilisées sur Internet. Ce sont les adresses privées suivantes en classe A, B et C :

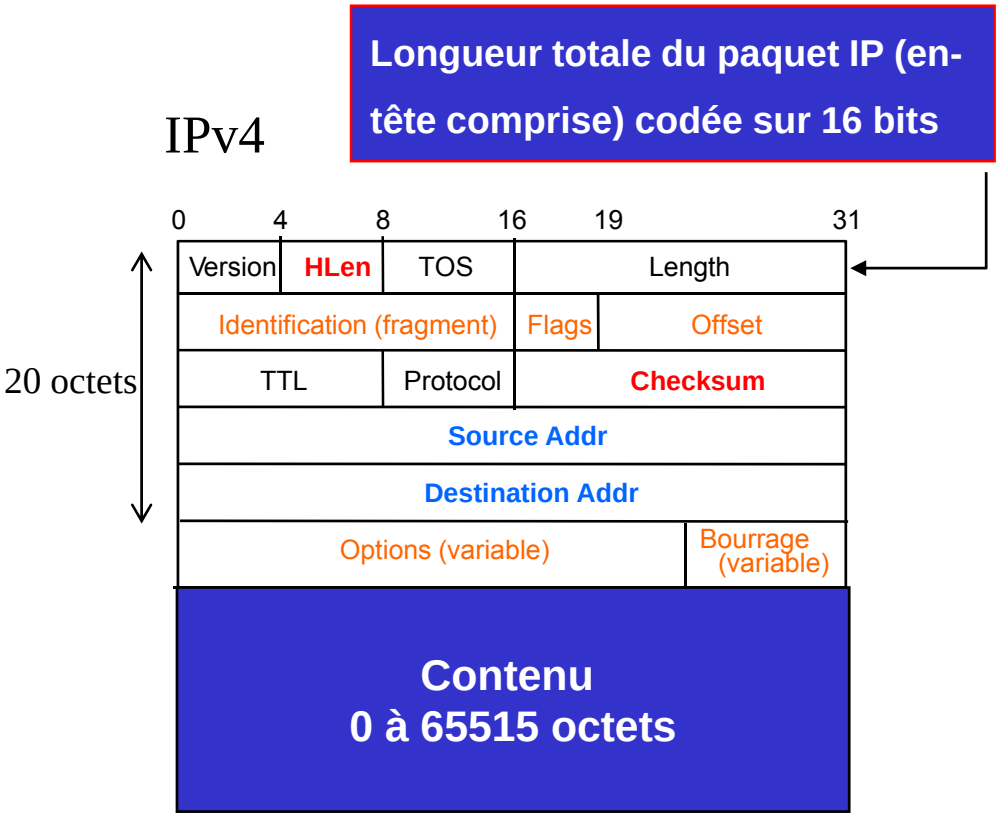
	Tranches d'adresses IP privées	Nombre de réseaux privés
10.0.0.0/8	10.0.0.0 à 10.255.255.255	1 réseau de classe A
172.16.0.0/12	172.16.0.0 à 172.31.255.255	16 réseaux de classe B
192.168.0.0/16	192.168.0.0 à 192.168.255.255.	256 réseaux de classe C

Adresses IP spéciales-1-

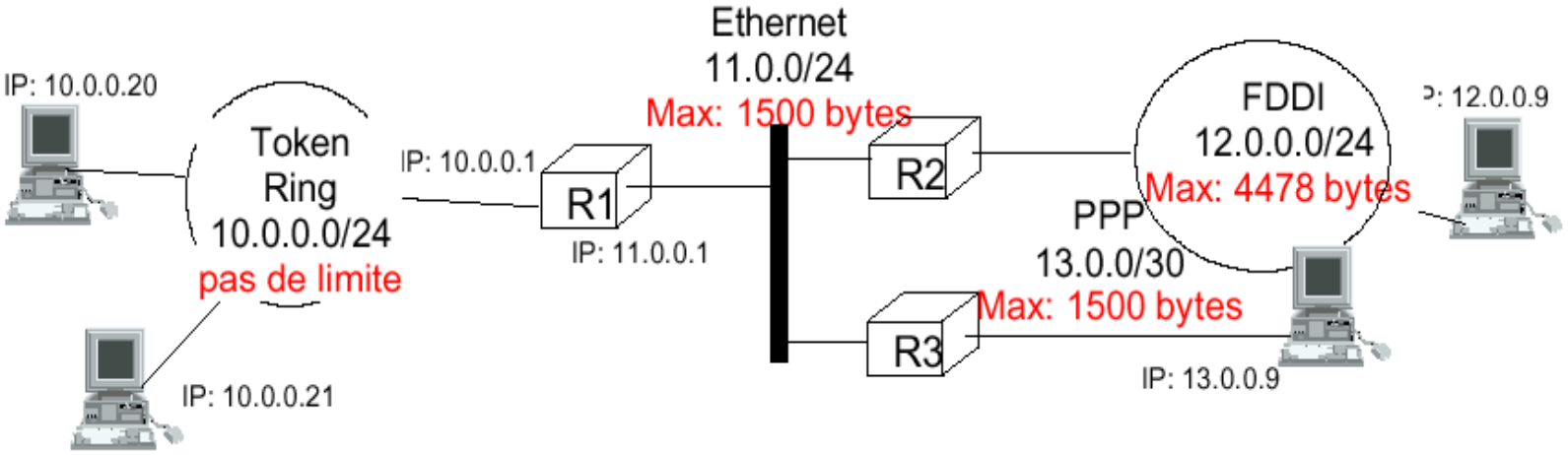


- Adresses spéciales : Les règles concernant les adresses IP prévoient un certain nombre d'adresses spéciales :
  - Adresses Réseaux : Dans ces adresses, la partie réservée à l'adresse station est à 0. Par exemple, 126.0.0.0 représente l'adresse réseau et non l'adresse d'un hôte.
  - Adresses Broadcast à diffusion dirigée : Dans ces adresses, la partie "adresse Station" a tous ses bits à 1. Par exemple, 126.255.255.255 est une adresse de broadcast sur le réseau 126. Les routeurs peuvent transmettre cette trame vers le réseau 126.
  - Adresses Broadcast à diffusion limitée. Dans ces adresses tous les bits sont à 1. (255.255.255.255). Cette trame est limitée au réseau de l'hôte qui l'envoie.
  - Adresses pour la maintenance ou adresses "Loopback" : 127.0.0.1 (Ping sur la station pour vérifier le fonctionnement de la pile IP locale).
  - Adresses réservées : Ce sont les adresses dont le numéro de réseau n'est composé que de 0 ou de 1.

Format des paquets IP



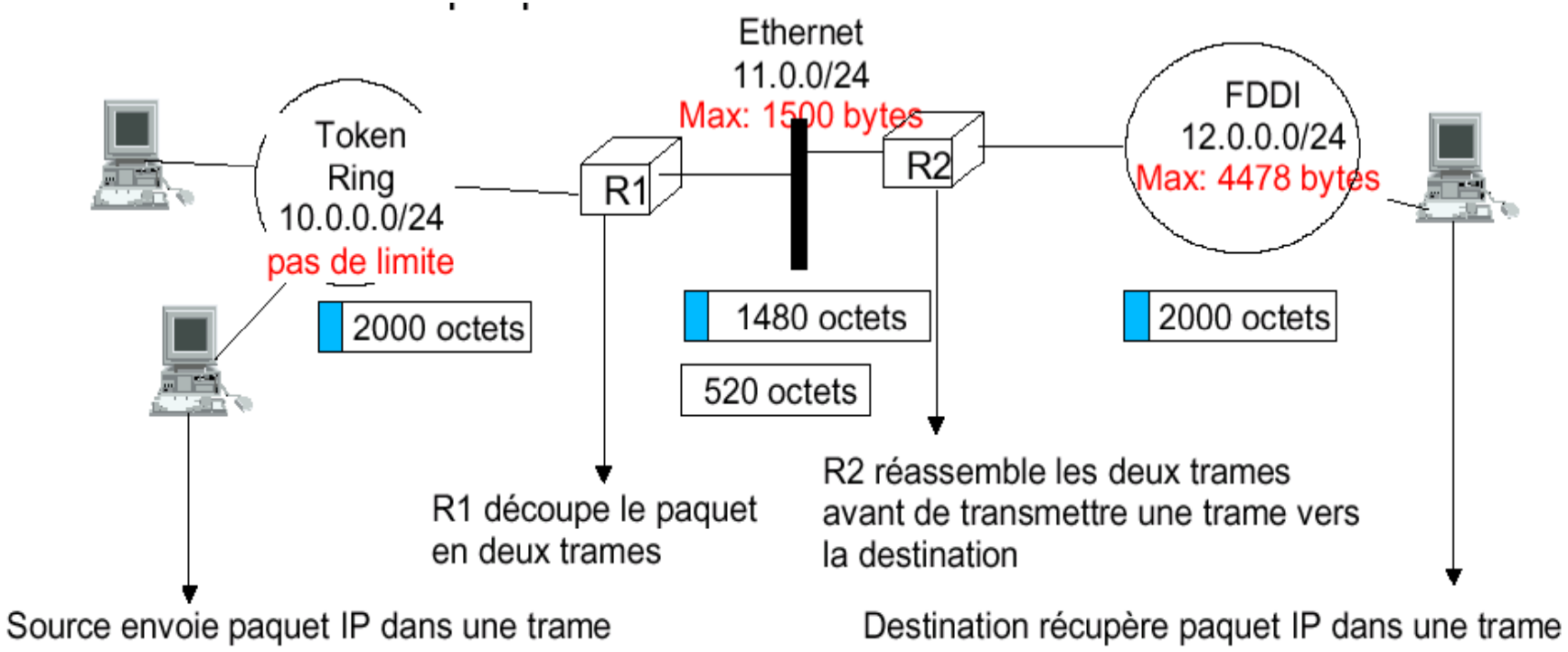
Pour transmettre l'information, il faut mettre le paquet IP à l'intérieure d'une trame



Problèmes :

1. IP supporte des paquets de 64 K octets au total
2. IP doit s'appuyer sur la couche liaison de données pour la transmission des paquets
3. La plupart des protocoles de la couche liaison de données ne supportent pas des trames aussi longues

- Idée : Permettre, au niveau de IP, le découpage des longs paquets sur chaque couche liaison de données si nécessaire
- Exemple : envoi de paquets IP contenant 2000 octets utiles

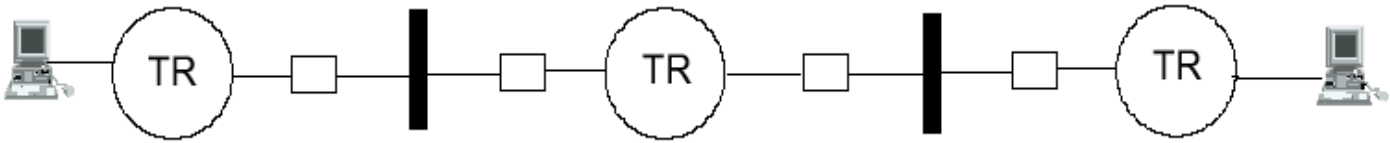


- Avantage :

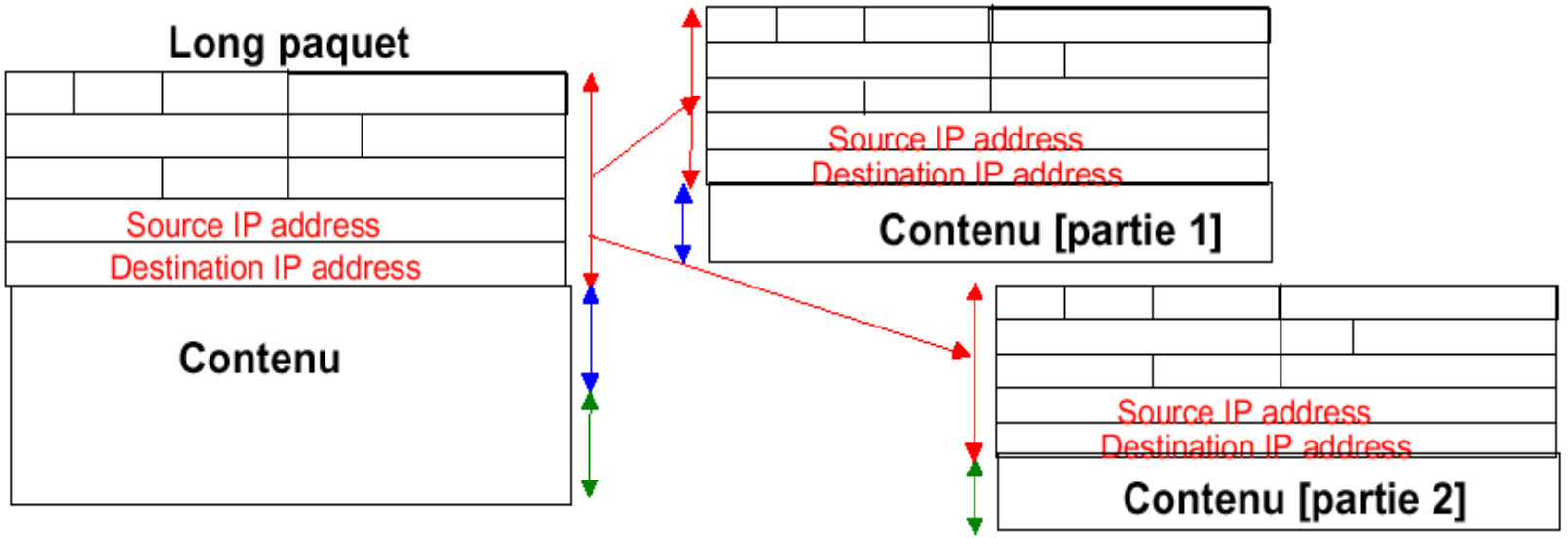
  bonne utilisation de chaque réseau intermédiaire

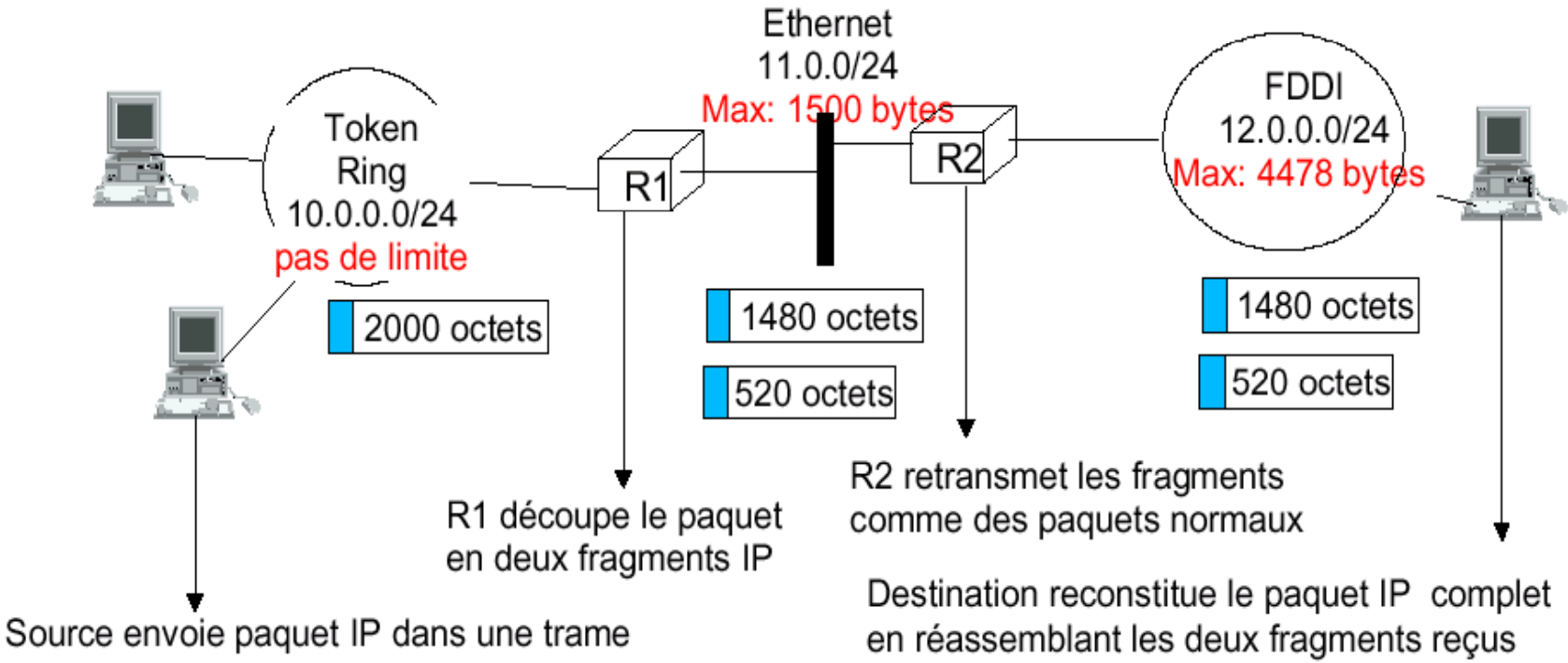
- Inconvénients :

  1. chaque station/routeur doit pouvoir fragmenter un paquet en N trames
  2. chaque station/routeur doit pouvoir réassembler N trames dans un paquet
     - Ce réassemblage consomme du CPU et de la mémoire
     - Ce réassemblage peut introduire un délai important
  3. un paquet peut devoir être *fragmenté plusieurs fois* avant d'arriver à destination



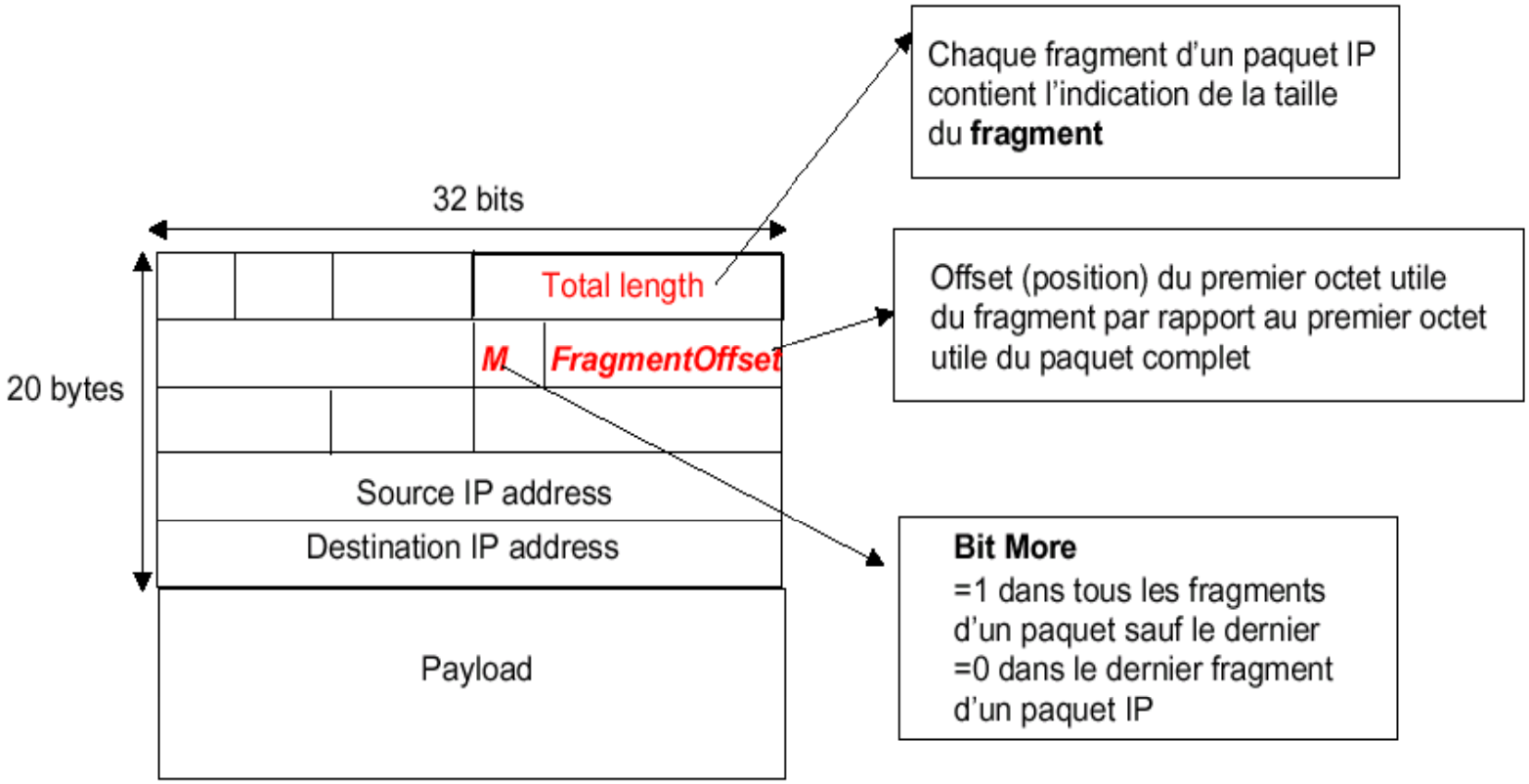
- **Amélioration**
  - Eviter de réassembler dans les routeurs (faciliter la vie des routeurs intermédiaires)
- **Principe**
  - Toute station et tout routeur est capable de fragmenter (découper) un paquet IP en fragments
    - Chaque fragment est un paquet IP contenant l'adresse IP source et l'adresse IP destination
  - Seule la destination effectue le réassemblage des fragments



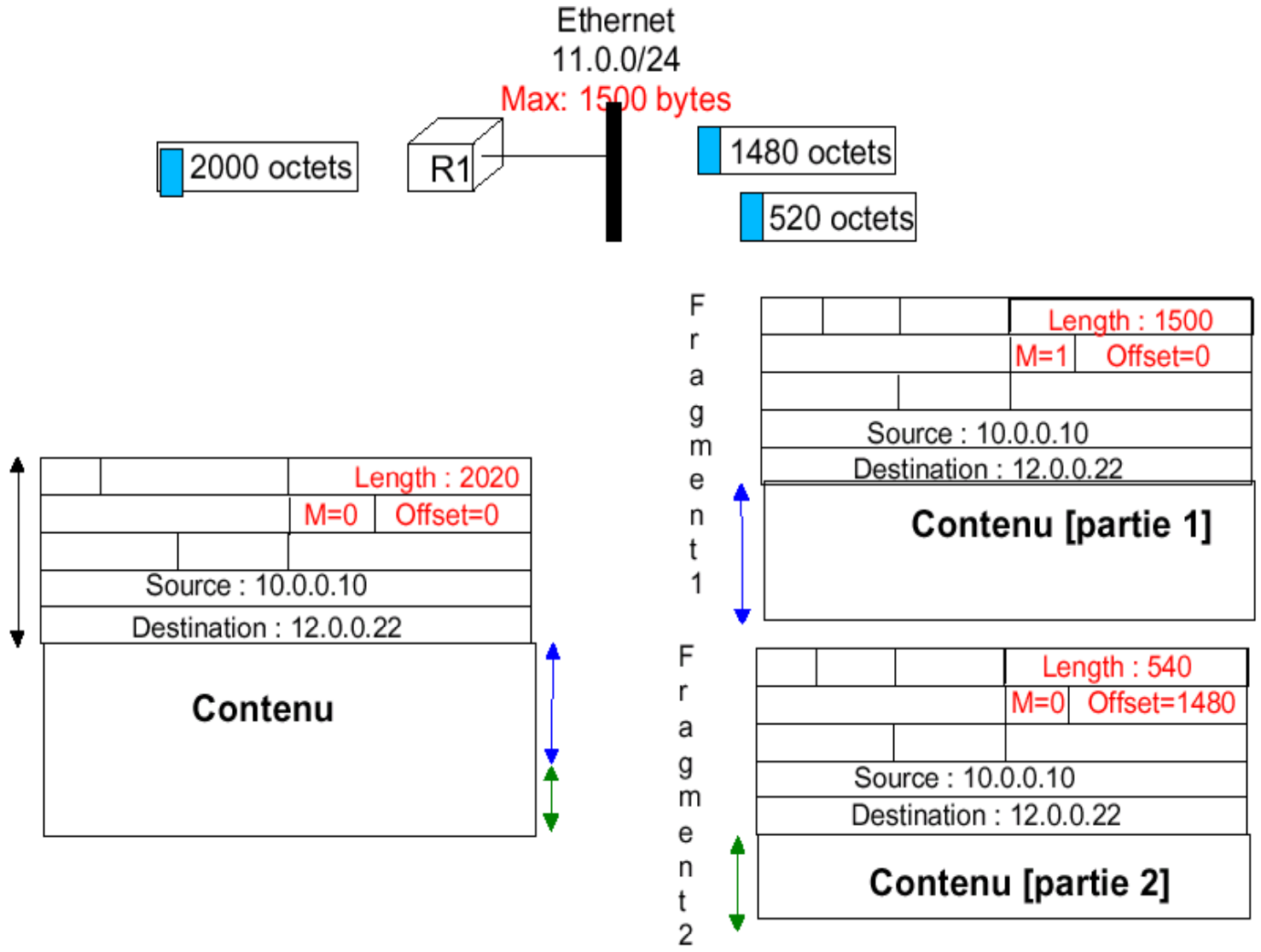


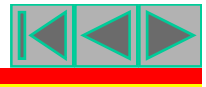
- Avantage : les routeurs n'effectuent plus le réassemblage
- Inconvénient : utilisation non optimale des réseaux intermédiaires (passage de deux paquets de petites taille plus tôt qu'un paquet de grande taille

- Principe :
  - Découper la partie contenu du paquet IP
  - Numéroter les fragments en cas de déséquencement



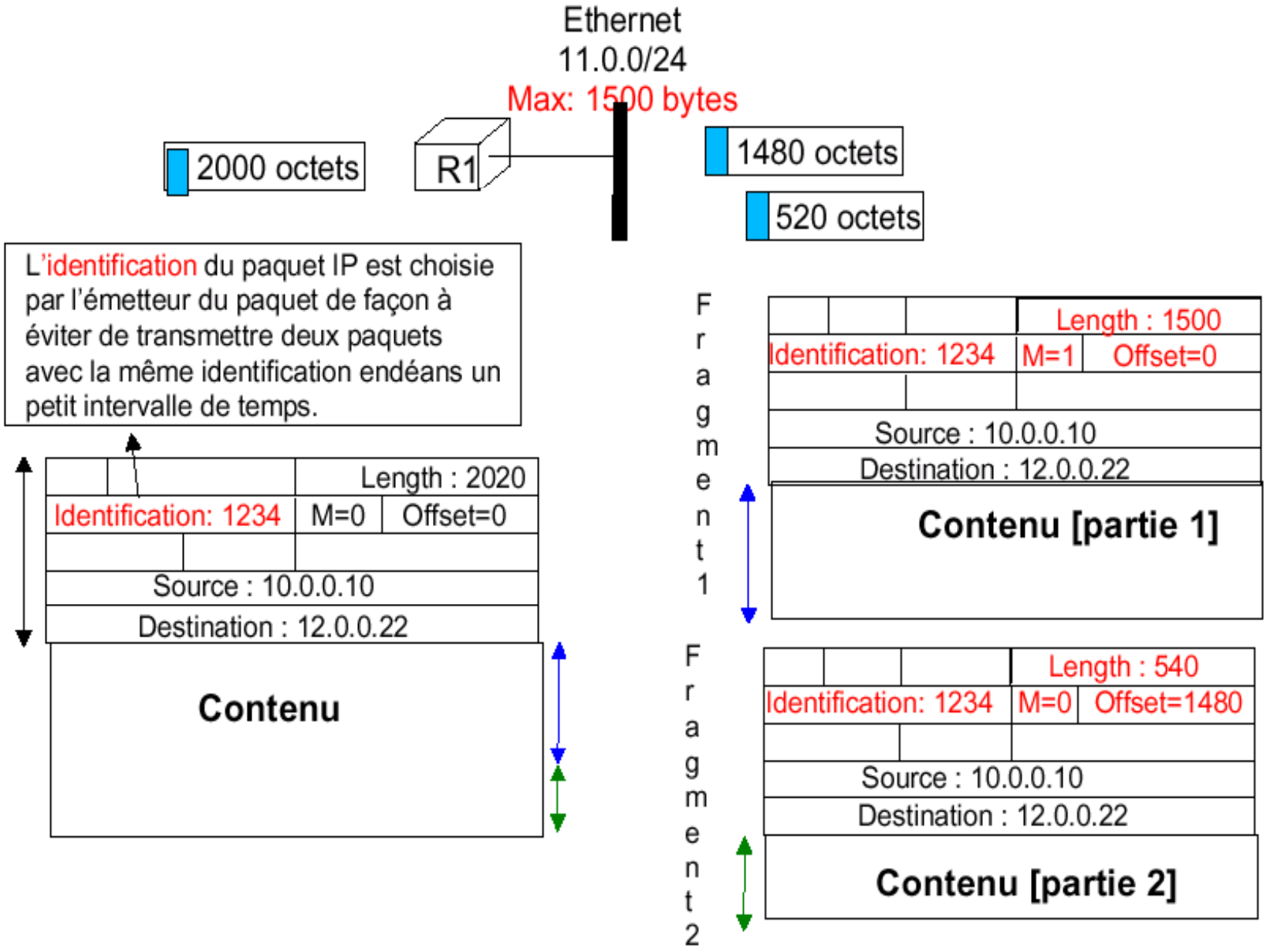
Fragmentation : exemple



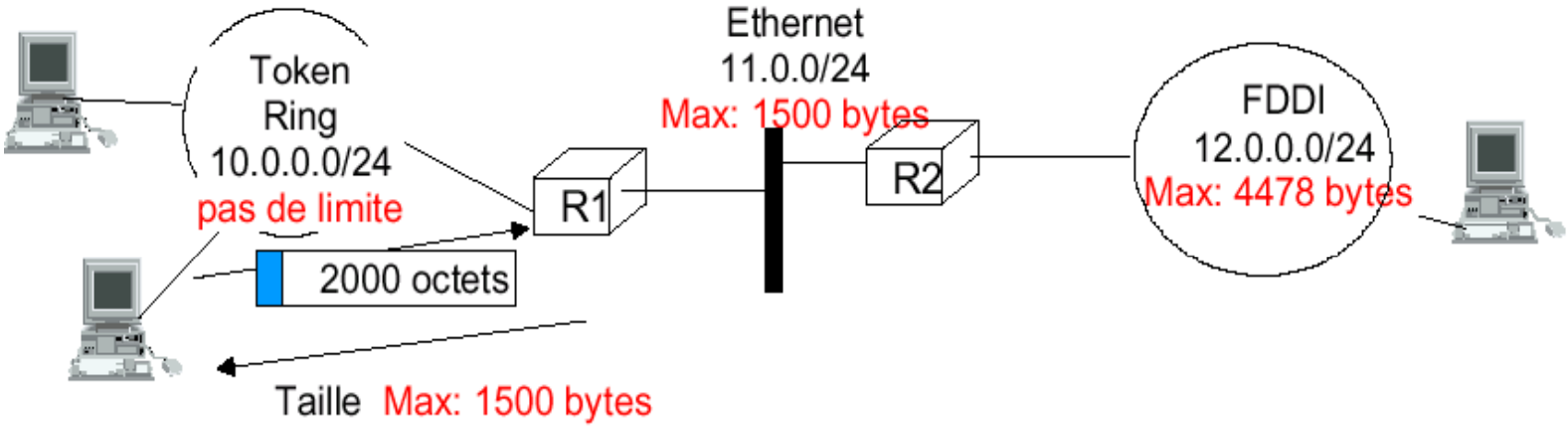


- Problèmes à résoudre :
  - Quand a-t-on reçu tous les fragments ?
    - Dernier fragment contient bit More=0
    - Que faire en cas de perte de fragments ?
      - dans ce cas, le paquet complet ne pourra pas être ré assemblé correctement par la destination (jeter le paquet)
  - Que faire en cas de déséquencement ?
    - Utiliser le champ Offset pour réordonner les fragments d'un même paquet IP
    - Comment faire face au désséquencement de plusieurs fragments provenant de paquets différents ?
      - Chacun fragment doit contenir **une identification du paquet** duquel il provient

Identification des paquets et fragments

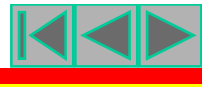


- Problème : Comment connaître la taille maximale des paquets à utiliser pour joindre une destination
- Solution : Plutôt que de fragmenter, un routeur peut informer la source de la taille maximale à utiliser



- Connaissant la taille maximale, l'utilisateur source peut adapter les paquets IP qu'il envoie en conséquence

IP : Problème des boucles

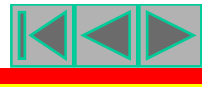


- Problème : Dans un réseau, des boucles peuvent se produire
  - Exemples
    - ✓ mauvaise configuration des tables de routage
    - ✓ pendant que les protocoles de routage distribuent une nouvelle route, les tables de routage peuvent être incohérentes
  - Si un paquet boucle entre quelques routeurs, il consomme inutilement de la bande passante

- Comment résoudre ce problème ?
  - Solution dans Ethernet : Spanning tree (arbre de recouvrement minimum)
- Remarques au sujet de STP:

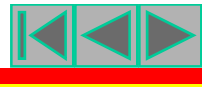
  → Certaines lignes du réseau sont inutilisées, gênant dans gros réseau (pas de transmission de trames de données durant le calcul du spanning tree)

  → Inutilisable dans un réseau IP



- Principe de la solution dans Internet :
  - ✓ chaque paquet contient un champ : Time-to-Live (TTL) indiquant le nombre maximum de routeurs intermédiaires que le paquet peut traverser
    exemple de valeur pour les nouveaux paquets : 64
  - ✓ chaque routeur vérifie le TTL des paquets reçus :
    Si TTL=1, supprimer le paquet en informant la source
    Si TTL>1, traiter le paquet et le transmettre vers la destination en décrémentant d'au moins une unité TTL

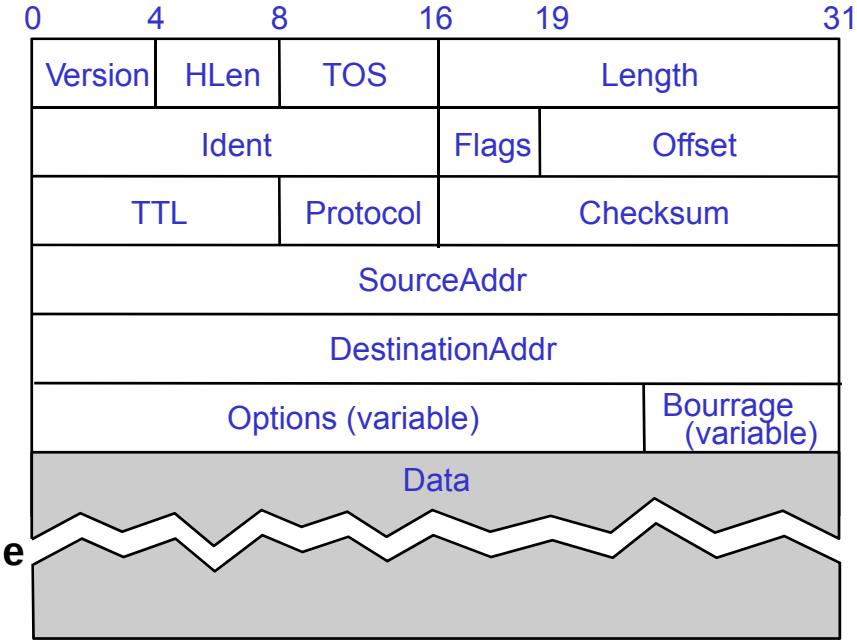
L'utilisation du TTL permet de limiter le temps de survie d'un paquet IP dans le réseau



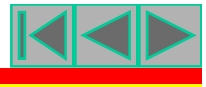
- Réaction face aux erreurs de transmission ?

1. Erreur de transmission sur le contenu du paquet
   - impact dépendra de l'application qui utilise ce paquet
   - IP : aucune détection pour erreurs sur le contenu (de ressort de l'application)
2. Erreur de transmission à l'intérieur de l'entête IP
   - potentiellement plus gênant
     - l'erreur peut modifier l'adresse source ou l'adresse destination
   - IP protège l'entête des paquets avec un checksum
     - ✓ somme de contrôle sur 16 bits calculé sur l'entête IP
     - ✓ tout routeur vérifie le checksum des paquets reçus et supprime tout paquet reçu avec un checksum erroné dans l'entête

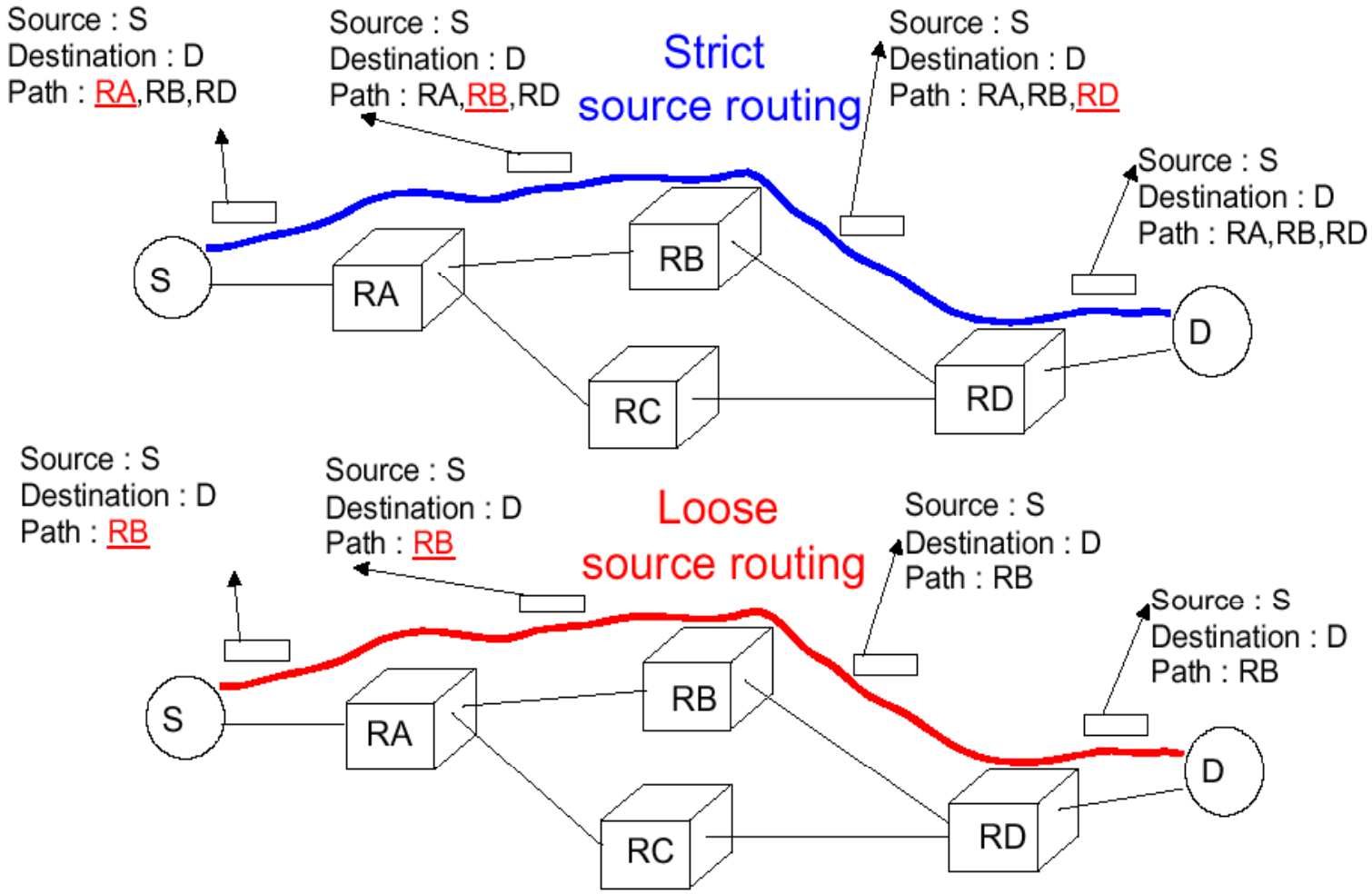
Format des paquets IP



- **Longueur: 20 octets – 60 octets**
  1. **Version : IPv4**
  2. **HLen: Longueur de l'en-tête**
  3. **TOS: Type of Service**
     - **Définition originale jamais utilisée**
     - **Base des Services Différenciés**
  4. **Longueur du datagramme (avec l'entête)**
     - **16 bit --> longueur max. 65535 octets**
     - **Nécessite la fragmentation en plusieurs trame**
       - **Entêtes Ident, Flags, Offset**
  5. **TTL: Time to live**
     - **Permet de capturer des paquets dans des boucles de routage**
  6. **Protocole**
     - **Désigne le protocole de la couche transport (normalement TCP ou UDP)**
  7. **Somme de contrôle**
     - **Protège l'en-tête IP, non pas les données**
     - **Calcul simple (différent de CRC) et protection moins forte**
  8. **Adresses source et destination : adresses des systèmes terminaux**
  9. **Options (p.ex. sécurité, options de routage) : Rarement utilisées**

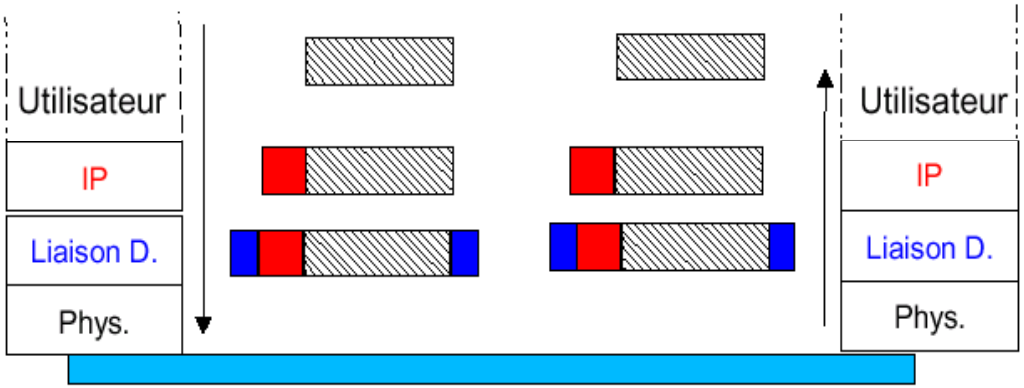


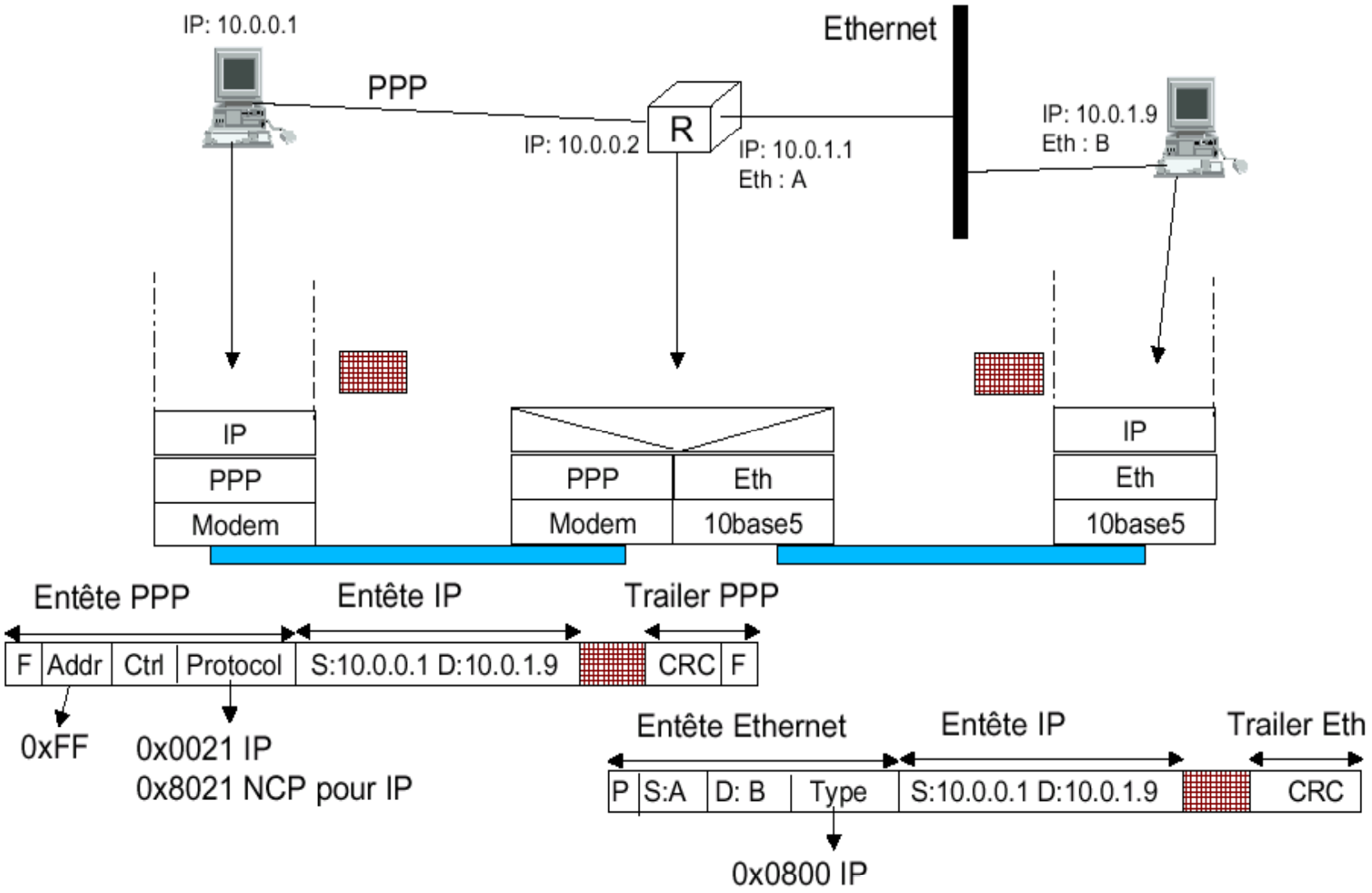
- Extensions possibles de l'entête IP :
  1. Strict source route option
     - permet à la source de spécifier de façon stricte la liste de tous les routeurs à utiliser pour atteindre la destination
  2. Loose « lâche : léger» source route option
     - permet à la source de spécifier la liste de certains routeurs intermédiaires à utiliser pour atteindre la destination
  3. Record route option
     - permet de demander à chaque routeur traversé par un paquet d'insérer son adresse dans les options (ça permet de connaître le chemin qui a été suivi pour arriver à la destination)
  4. Router alert
     - Permet d'indiquer aux routeurs intermédiaires qu'ils doivent faire attention en traitant ce paquet particulier
- Contrainte : maximum 64 octets pour entête + options (44 octets ⇔ 11 adresses IP)

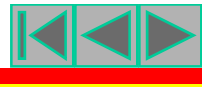

Source : S
Destination : D
Path : RA,RB,RD
Source : S
Destination : D
Path : RA,RB,RD
Strict
source routing
Source : S
Destination : D
Path : RA,RB,RD
Source : S
Destination : D
Path : RA,RB,RD
S
RA
RB
RC
RD
D
Source : S
Destination : D
Path : RB
Source : S
Destination : D
Path : RB
Loose
source routing
Source : S
Destination : D
Path : RB
Source : S
Destination : D
Path : RB
S
RA
RB
RC
RD
D

1. Les données utilisateurs vont être traités par la couche IP,
2. La couche IP va rajouter une entête spécifique de façon a en constituer un paquet
3. Ce paquet va être fourni à la couche liaison de données qui va avoir pour objectif de transmettre le paquet sur le réseau vers la destination

   - Pour la transmission, la couche liaison de données va construire une trame qui contient **AS** et **AD** et un contrôle (CRC)
   - La destination, au niveau de la couche liaison de données , va vérifier le CRC, l'AD puis il va fournir le contenu de la trame en s'appuyant dans le cas d'Internet sur le champ type qui indique qui est l'utilisateur du paquet IP

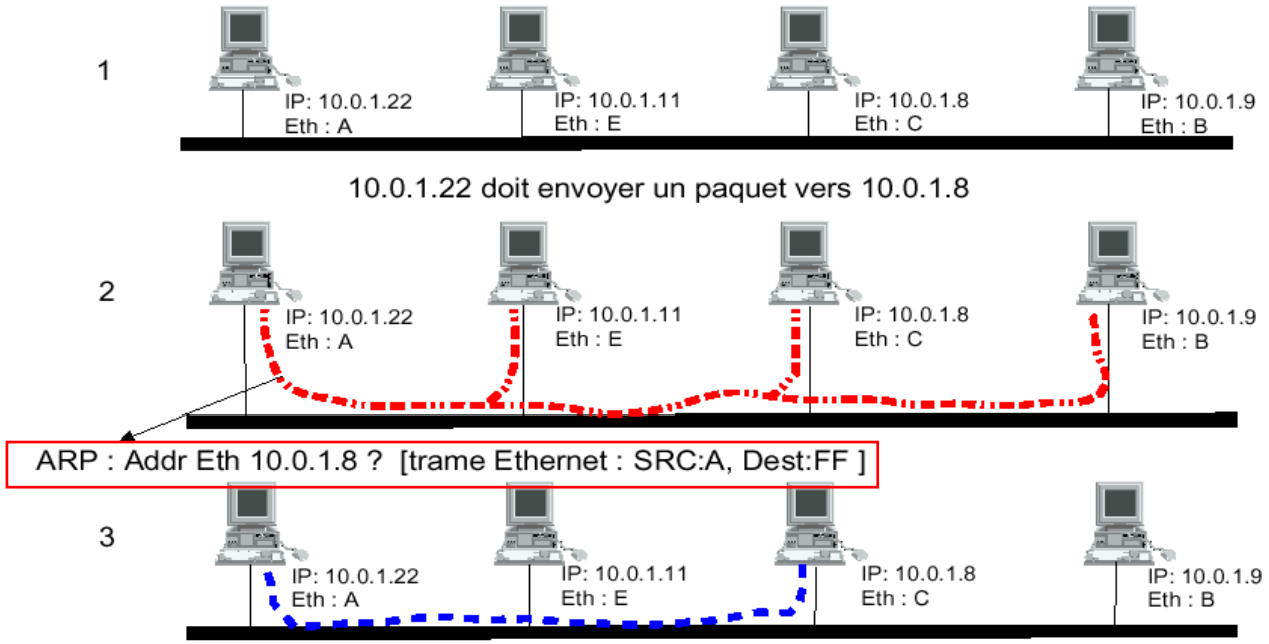



IP: 10.0.0.1
PPP
IP: 10.0.0.2
R
IP: 10.0.1.1
Eth : A
Ethernet
IP: 10.0.1.9
Eth : B
IP
PPP
Modem
PPP
Eth
Modem
10base5
IP
Eth
10base5
Entête PPP
Entête IP
Trailer PPP
F
Addr
Ctrl
Protocol
S:10.0.0.1 D:10.0.1.9
CRC
F
0xFF
0x0021 IP
0x8021 NCP pour IP
Entête Ethernet
Entête IP
Trailer Eth
P
S:A
D: B
Type
S:10.0.0.1 D:10.0.1.9
CRC
0x0800 IP

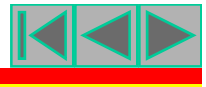


- Dans un réseau local, comment trouver l'adresse MAC qui permet de joindre une adresse IP donnée ?

1. LAN supporte l'envoi de trames broadcast
   - envoyer une trame broadcast demandant à quelle adresse de couche 2 une adresse IP correspond
   - la station ayant cette adresse IP répondra
     - solution utilisée pour IP sur Ethernet (protocole ARP)
2. LAN ne supporte pas l'envoi de trames broadcast.
   - placer sur serveur les couples *adresse IP*:*adresse MAC*
   - contacter le serveur pour connaître une adresse MAC
     - solution utilisée dans les réseaux de type ATM



- **Première étape pour : une station cherche à avoir l'adresse MAC de la station de destination**
  - Envoie une trame Broadcast (qui sera reçu par toutes les stations)
  - La station concernée va répondre en lui indiquant l'adresse MAC qu'il faut utiliser pour lui joindre
  - La station source pourrai connaître l'adresse MAC qu'il faut utiliser ce qui lui permet de construire sa trame de liaison de données
- **Exemple ci-dessus**
  - La station C (Adr IP : 10.0.1.8) répond à A en envoyant dans une trame Ethernet son adresse Ethernet
    - Trame Ethernet construite par C : SRC:C Dest : A
    - Sur base de cette réponse, 10.0.1.22 pourra facilement envoyer son paquet



- Faut-il envoyer une requête ARP pour chaque paquet IP ?

  Réponse : Non

  - chaque station conserve une table ARP reprenant la correspondance entre adresse IP et adresse Ethernet
  - ARP est utilisé lorsque l'info n'est pas dans la table

- Comment faire si une station change d'adresse Ethernet ?

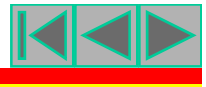
  Réponse :

  Temporisateur associée à chaque entrée de la table ARP

  ⇨ Temporisateur expire : entrée supprimée (durée 15 mn)

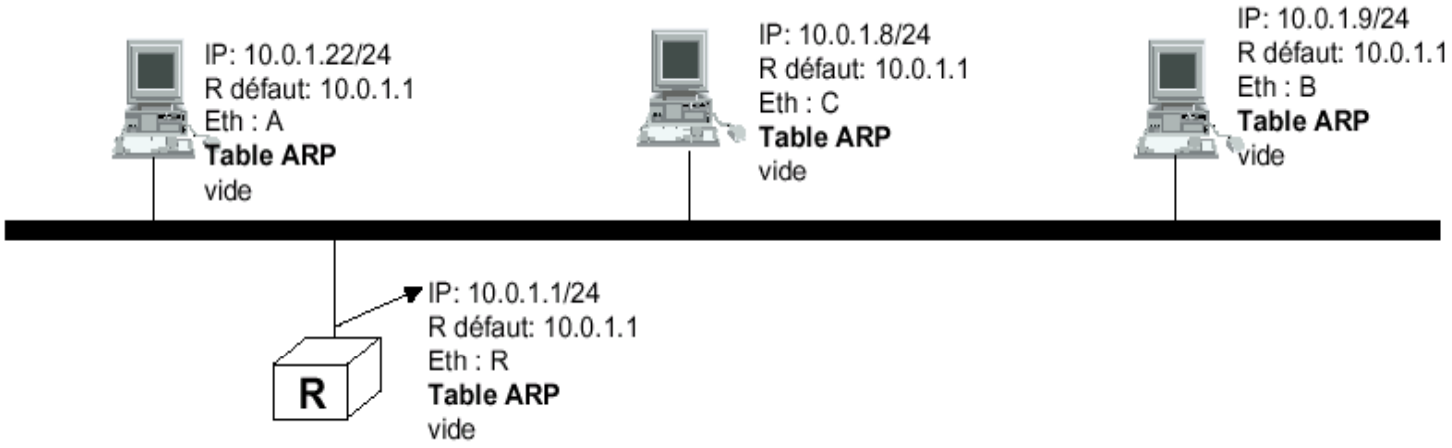
  ⇨ Sur validation d'une ligne de la table :

  ➔ Initialisation du temporisateur



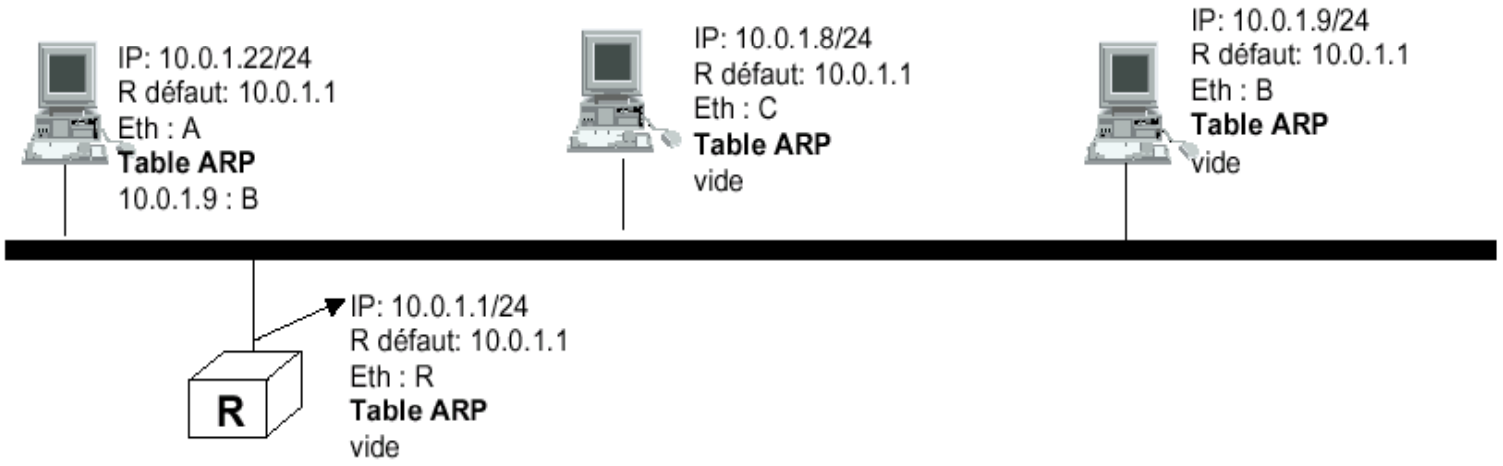
- Informations nécessaires à une station IP (pour qu'elle puisse fonctionner)

1. adresses MAC des interfaces de la station
2. adresses IP des interfaces de la station
   - Pour chaque adresse IP, le masque de sous-réseau permet de connaître les adresses IP joignables via le réseau local auquel on est connecté
3. table de routage : contient trois types d'information
   1. réseaux directement connectés
   2. réseaux connus de la station (spécifié soit par configuration, soit par configuration ou grâce au protocole ICMP)
   3. routeur par défaut
      - route vers le réseau 0.0.0.0/0 (masque = 0 ⇔ ensemble des adresses IP)



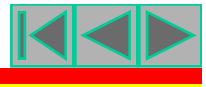
A. Envoi d'un paquet IP de 10.0.1.22 vers 10.0.1.9

1. Vérifier si l'adresse destination se trouve dans le sous réseau auquel la station est directement connectée
2. oui, il faut transmettre le paquet directement sur l'Ethernet
3. consulter la table ARP pour obtenir l'adresse Eth pour 10.0.1.9
4. adresse Eth pas dans table -> envoi trame ARP en broadcast
5. trame ARP de réponse venant de Eth:B (10.0.1.9<=>Eth:B)
6. mise à jour de la table ARP
7. envoi du paquet IP (S:10.0.1.22 D:10.0.1.9) dans trame Eth
   - source : Eth:A, destination: Eth:B



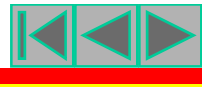
B. Envoi d'un paquet IP de 10.0.1.22 vers 10.0.2.9

1. vérifier si l'adresse destination se trouve dans le sous réseau auquel la station est directement connectée
2. Réponse :Non, il faut, alors transmettre le paquet au routeur par défaut
   - routeur par défaut (10.0.1.1) est connecté à l'Ethernet local
3. consulter la table ARP pour obtenir l'adresse Eth pour 10.0.1.1
4. adresse Eth n'est pas dans table ⇨ envoi trame ARP en broadcast
5. trame ARP de réponse venant de Eth:R (10.0.1.1 ⇔ Eth:R)
6. Mise à jour de la table ARP
7. envoi du paquet IP (S:10.0.1.22, D:10.0.2.9) dans trame Ethernet
   - source : Eth:A, destination: Eth:R



Que doit connaître un routeur IP ?

1. Adresses MAC des interfaces du routeur
2. Adresses IP des interfaces du routeur
   - pour chaque adresse IP, le masque de sous-réseau permet de connaître les adresses IP joignables via le LAN
3. Table de routage
   1. réseaux directement connectés
   2. réseaux connus du routeur
      - Soit par configuration
      - Soit grâce aux protocoles de routage
   3. éventuellement routeur par défaut
      1. route vers le réseau 0.0.0.0/0



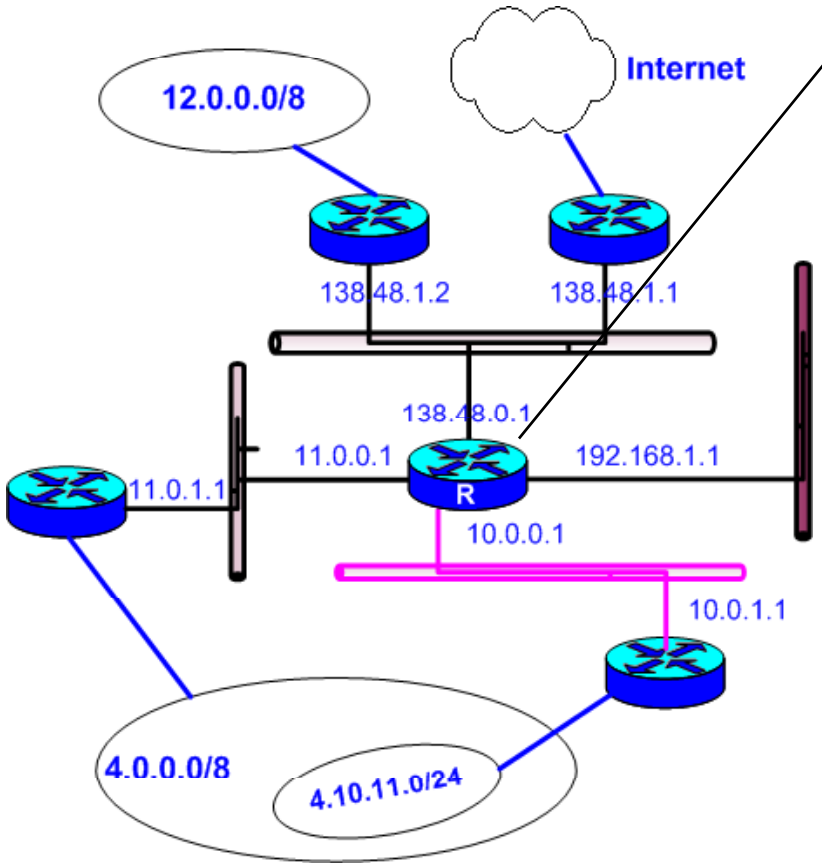
Arrivée d'un paquet IP :

1. Vérifier si la destination du paquet est une des adresses IP du routeur
   - si oui, le paquet est arrivé à destination
2. Vérifier si le paquet est destiné à un des réseaux auxquels le routeur est directement connecté
   - comparer les M premiers bits de l'adresse destination du paquet avec M premiers bits de chaque réseau
     - transmettre le paquet sur ce réseau
3. si le paquet ne concerne pas un réseau auquel le routeur est directement connecté

Alors : Rechercher dans la table de routage

⇒Transmettre ce paquet sur la route la plus spécifique (qui a le masque le plus long)

- comment ?
  - Comparer les M premiers bits de l'adresse destination avec les M premiers bits des routes pour trouver correspondance la plus longue (spécifique)
    - transmettre le paquet en suivant cette route



Adresses IP
11.0.0.1 ; 138.48.0.1 ; 192.168.1.1 ; 10.0.0.1

Table de routage

Réseaux directement connectés

138.48.0.0/16	[Nord]
11.0.0.0/8	[Ouest]
192.168.1.0/24	[Est]
10.0.0.0/8	[Sud]

Réseaux accessibles via routeurs intermédiaires

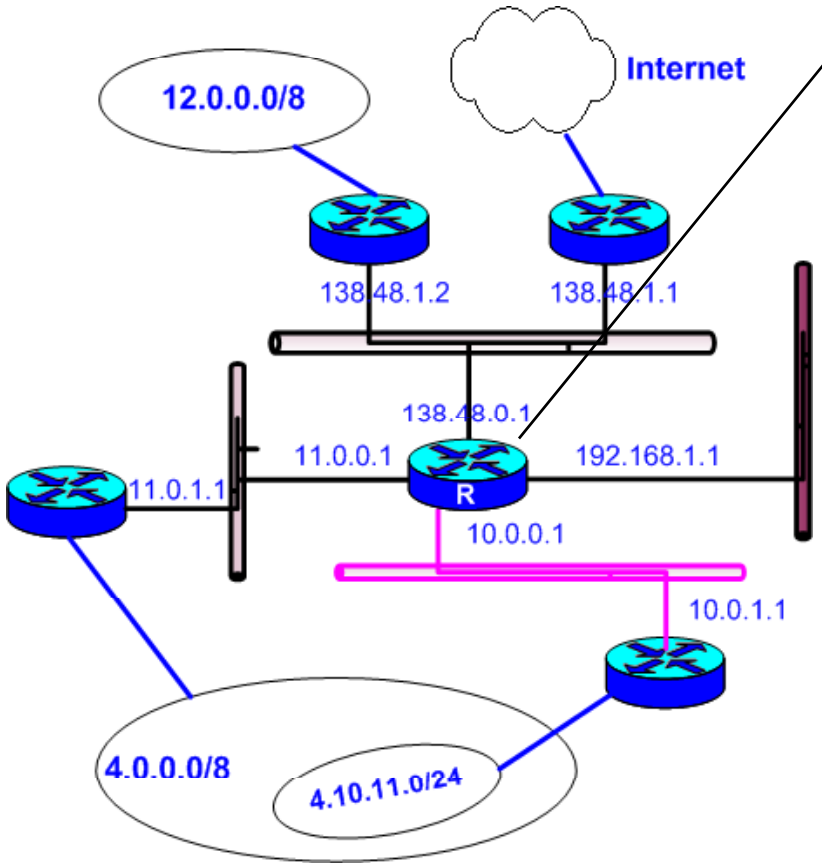
4.0.0.0/8	via 11.0.1.1	[Ouest]
4.10.11.0/24	via 10.0.1.1	[Sud]
12.0.0.0/8	via 138.48.1.2	[Nord]
0.0.0.0/0	via 138.48.1.1	[Nord]

Cas N°1 : Arrivée en R d'un paquet avec Dest:192.168.1.1

Si (dest=adresse du routeur) alors

paquet arrivé à sa destination /* dans ce cas OUI*/

Fonctionnement d'un routeur IP- 4-



Adresses IP
11.0.0.1 ; 138.48.0.1 ; 192.168.1.1 ; 10.0.0.1

Table de routage

Réseaux directement connectés

138.48.0.0/16	[Nord]
11.0.0.0/8	[Ouest]
192.168.1.0/24	[Est]
10.0.0.0/8	[Sud]

Réseaux accessibles via routeurs intermédiaires

4.0.0.0/8	via 11.0.1.1	[Ouest]
4.10.11.0/24	via 10.0.1.1	[Sud]
12.0.0.0/8	via 138.48.1.2	[Nord]
0.0.0.0/0	via 138.48.1.1	[Nord]

Arrivée en R d'un paquet avec Dest::138.48.99.17

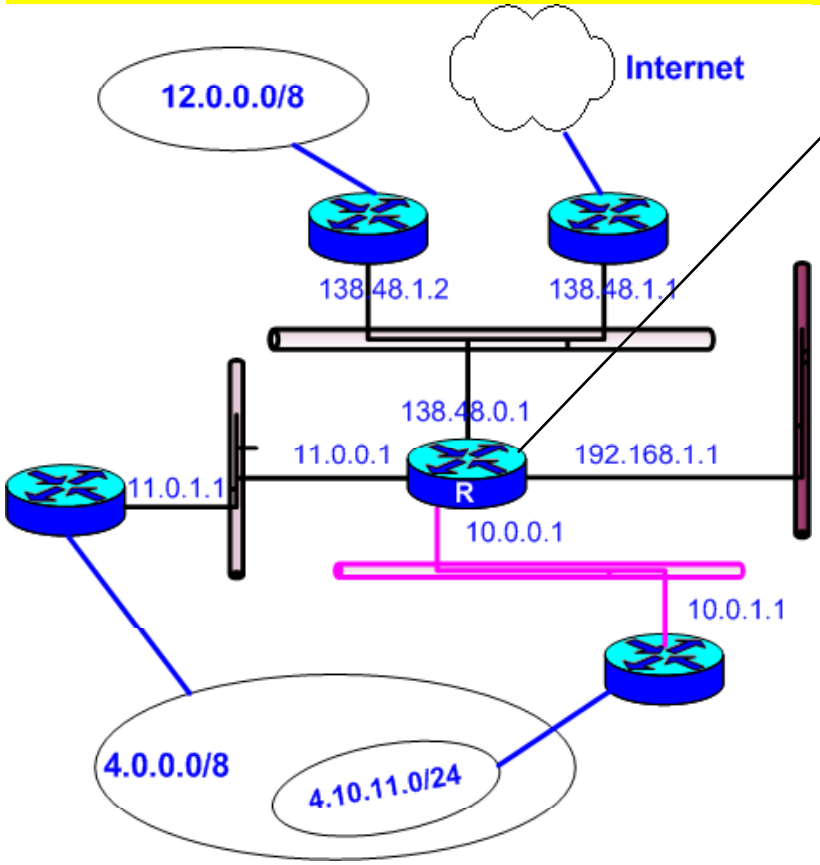
Dest = adresse du routeur ?

Non

dest dans réseau directement connecté ?

Oui, envoyer le paquet sur Ethernet Nord

Fonctionnement d'un routeur IP- 5-



Adresses IP
11.0.0.1 ; 138.48.0.1 ; 192.168.1.1 ; 10.0.0.1

Table de routage

Réseaux directement connectés

138.48.0.0/16	[Nord]
11.0.0.0/8	[Ouest]
192.168.1.0/24	[Est]
10.0.0.0/8	[Sud]

Réseaux accessibles via routeurs intermédiaires

4.0.0.0/8	via 11.0.1.1	[Ouest]
4.10.11.0/24	via 10.0.1.1	[Sud]
12.0.0.0/8	via 138.48.1.2	[Nord]
0.0.0.0/0	via 138.48.1.1	[Nord]

Adr destination : 4.92.18.43

Routes candidates : (comparaison ligne par ligne)

Adr : 4.0.0.0/8 : Le 8 premiers bits sont identiques => est une route candidate

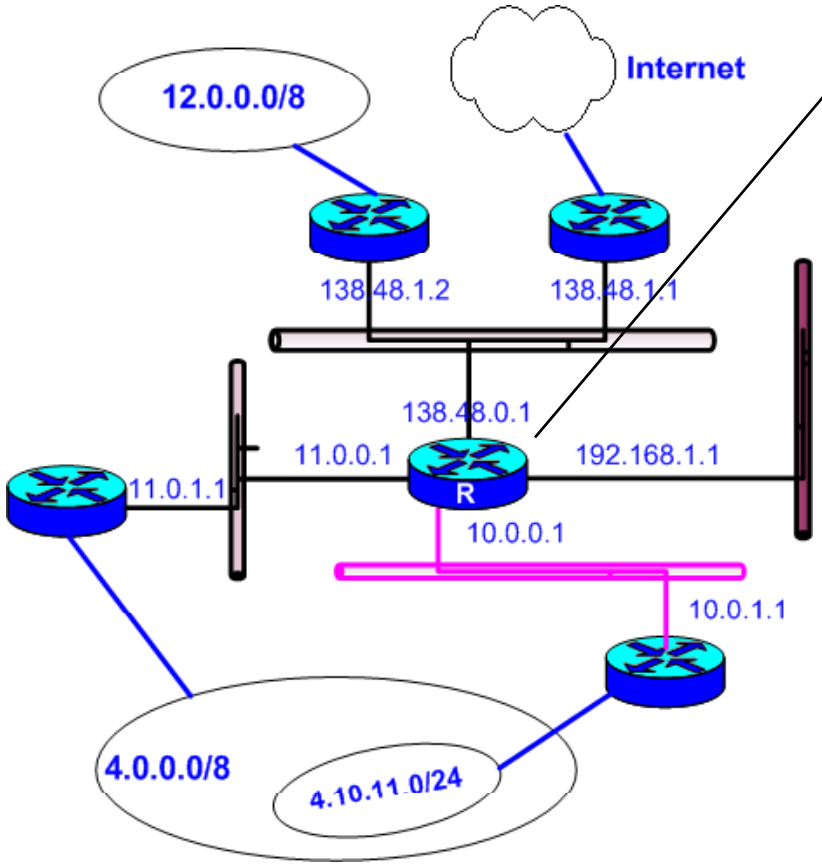
Adr : 4.10.11.0/24 pas de correspondance (cette route n'est pas candidate)

Adr 12.0.0.0 /8 pas de correspondance (cette route n'est pas candidate)

Adr : 0.0.0.0 /0 on va comparer rien de tout avec 0, il y correspondance (cette route est candidate)

L'adresse qui a le masque le plus long est celle : 4.0.0.0/8

Fonctionnement d'un routeur IP- 6-



Adresses IP
11.0.0.1 ; 138.48.0.1 ; 192.168.1.1 ; 10.0.0.1

Table de routage

Réseaux directement connectés

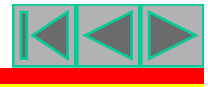
138.48.0.0/16	[Nord]
11.0.0.0/8	[Ouest]
192.168.1.0/24	[Est]
10.0.0.0/8	[Sud]

Réseaux accessibles via routeurs intermédiaires

4.0.0.0/8	via 11.0.1.1	[Ouest]
4.10.11.0/24	via 10.0.1.1	[Sud]
12.0.0.0/8	via 138.48.1.2	[Nord]
0.0.0.0/0	via 138.48.1.1	[Nord]

Arrivée en R d'un paquet avec Dest::4.10.11.199

- Dest = adresse du routeur ? Non
- dest dans réseau directement connecté ? Non
- Trouver dans table de routage route la plus spécifique
  - route vers 4.10.11.0/24 via 10.0.1.1 [Sud]
  - envoyer le paquet sur port Sud



- Forwarding:
  - consiste à sélectionner un port de sortie sur la base de l'adresse de destination et la table de routage
  - En général, implémentation hardware
- Routage :
  - le processus par lequel, les tables de routage sont construites
  - Emplémenté en logiciel
  - Essayer la commande : "netstat –rn"

Table de routage

Adresses IP
11.0.0.1 ; 138.48.0.1 ; 192.168.1.1 ; 10.0.0.1

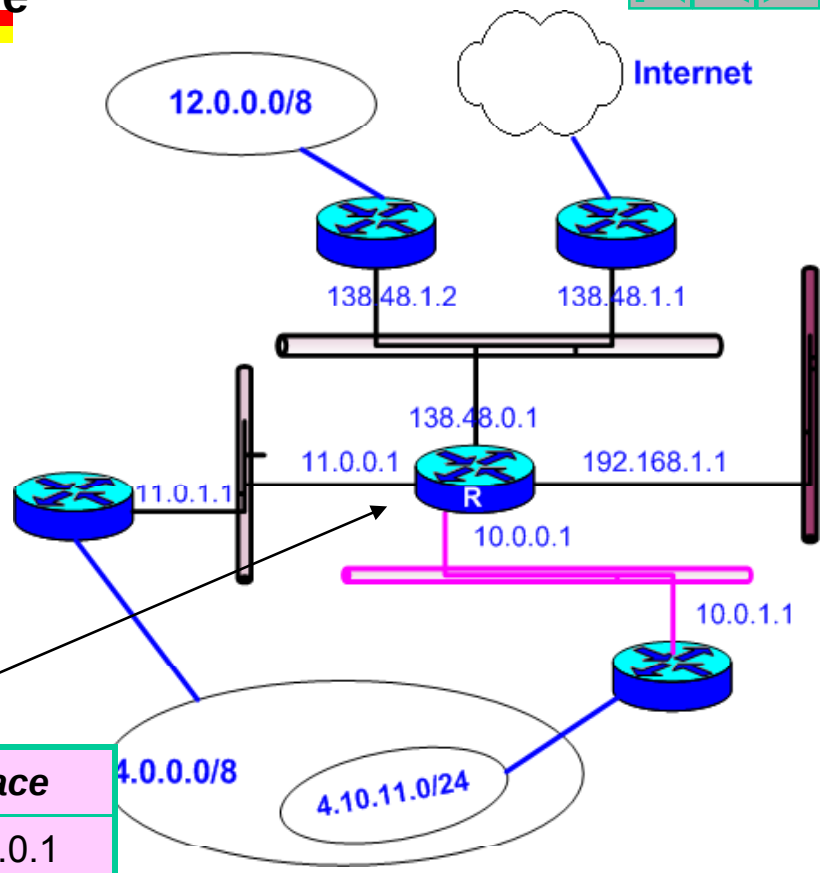
Table de routage

Réseaux directement connectés

138.48.0.0/16 [Nord]
11.0.0.0/8 [Ouest]
192.168.1.0/24 [Est]
10.0.0.0/8 [Sud]

Réseaux accessibles via routeurs intermédiaires

4.0.0.0/8 via 11.0.1.1 [Ouest]
4.10.11.0/24 via 10.0.1.1 [Sud]
12.0.0.0/8 via 138.48.1.2 [Nord]
0.0.0.0/0 via 138.48.1.1 [Nord]



Déstination	*Next Hop*	*Métriqe*	*Interface*
138.48.0.0/16	**Direct**	**0**	138.48.0.1
11.0.0.0/8	**Direct**	**0**	11.0.0.1
192.168.1.0/24	**Direct**	**0**	192.168.1.1
10.0.0.0/8	**Direct**	**0**	10.0.0.1
4.0.0.0/8	11.0.1.1	-	11.0.0.1
4.10.11.0/24	10.0.1.1	-	10.0.0.1
12.0.0.0/8	138.48.1.2	-	138.48.0.1
0.0.0.0/0	138.48.1.1	-	138.48.0.1

Autre Exemple de Table de routage

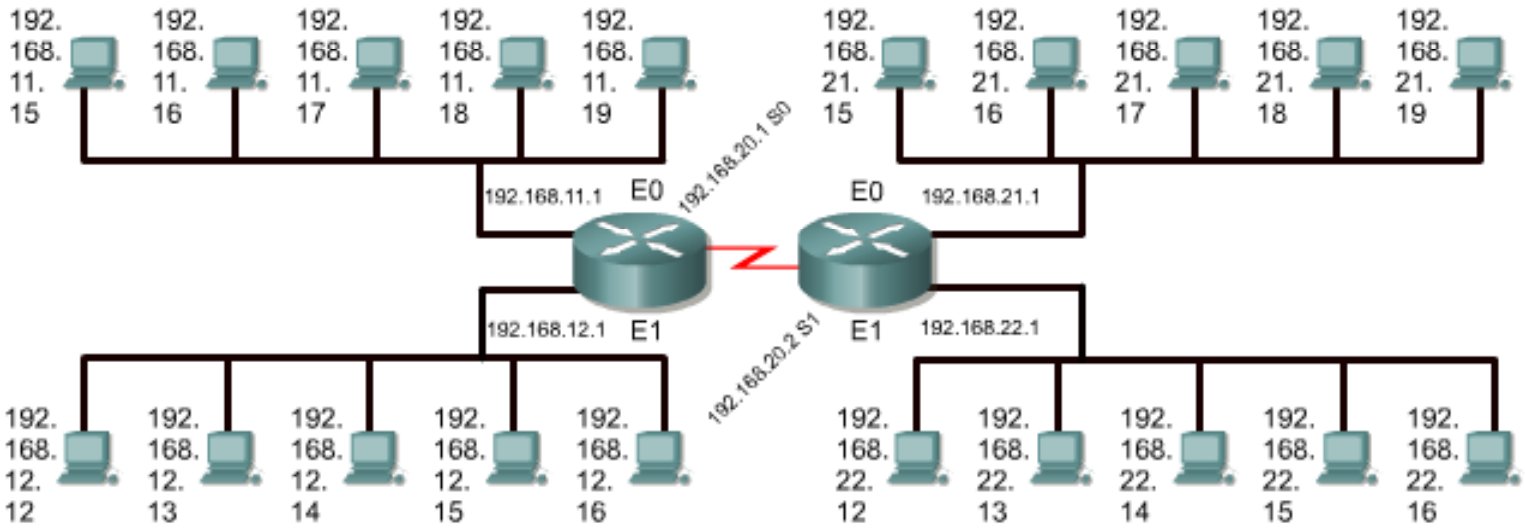


Table de routage

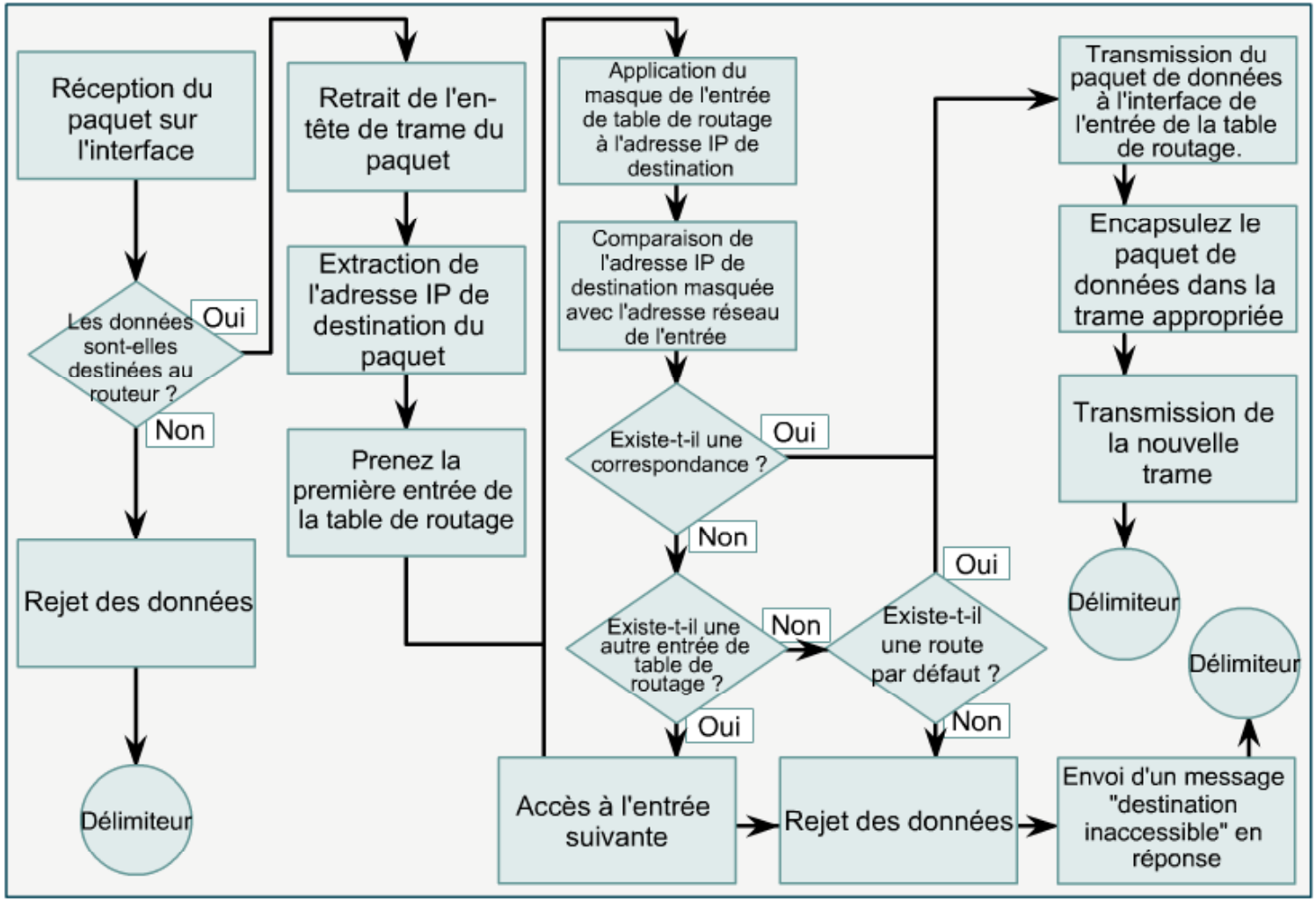
Appris	Adresse réseau	Saut	Interface
C -	192.168.11.0	0	E0
C -	192.168.12.0	0	E1
C -	192.168.20.0	0	S0
R -	192.168.21.0	1	S0
R -	192.168.22.0	1	S0

Table de routage

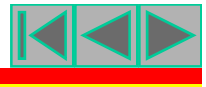
Appris	Adresse réseau	Saut	Interface
C -	192.168.21.0	0	E0
C -	192.168.22.0	0	E1
C -	192.168.20.0	0	S1
R -	192.168.11.0	1	S1
R -	192.168.12.0	1	S1

1. ***Type de protocole:*** *cette information identifie le type de protocole de routage qui a créé chaque entrée.*
2. ***Associations du saut suivant:*** *indique au routeur que la destination lui est directement connectée, ou qu'elle peut être atteinte par le biais d'un autre routeur appelé le «saut suivant» vers la destination finale. Dès réception d'un paquet, le routeur vérifie l'adresse de destination et tente de trouver une correspondance dans sa table de routage.*
3. ***Métrique de routage:*** *les métriques utilisées varient selon les protocoles de routage et permettent de déterminer les avantages d'une route sur une autre. Par exemple, le protocole RIP se sert d'une seule métrique de routage : le nombre de sauts. Le protocole IGRP crée une valeur de métrique composite à partir des métriques de fiabilité, de délai, de charge et de bande passante.*
4. ***Interfaces de sortie****: cette information désigne l'interface à partir de laquelle les données doivent être envoyées pour atteindre leur destination finale.*

Détermination du chemin



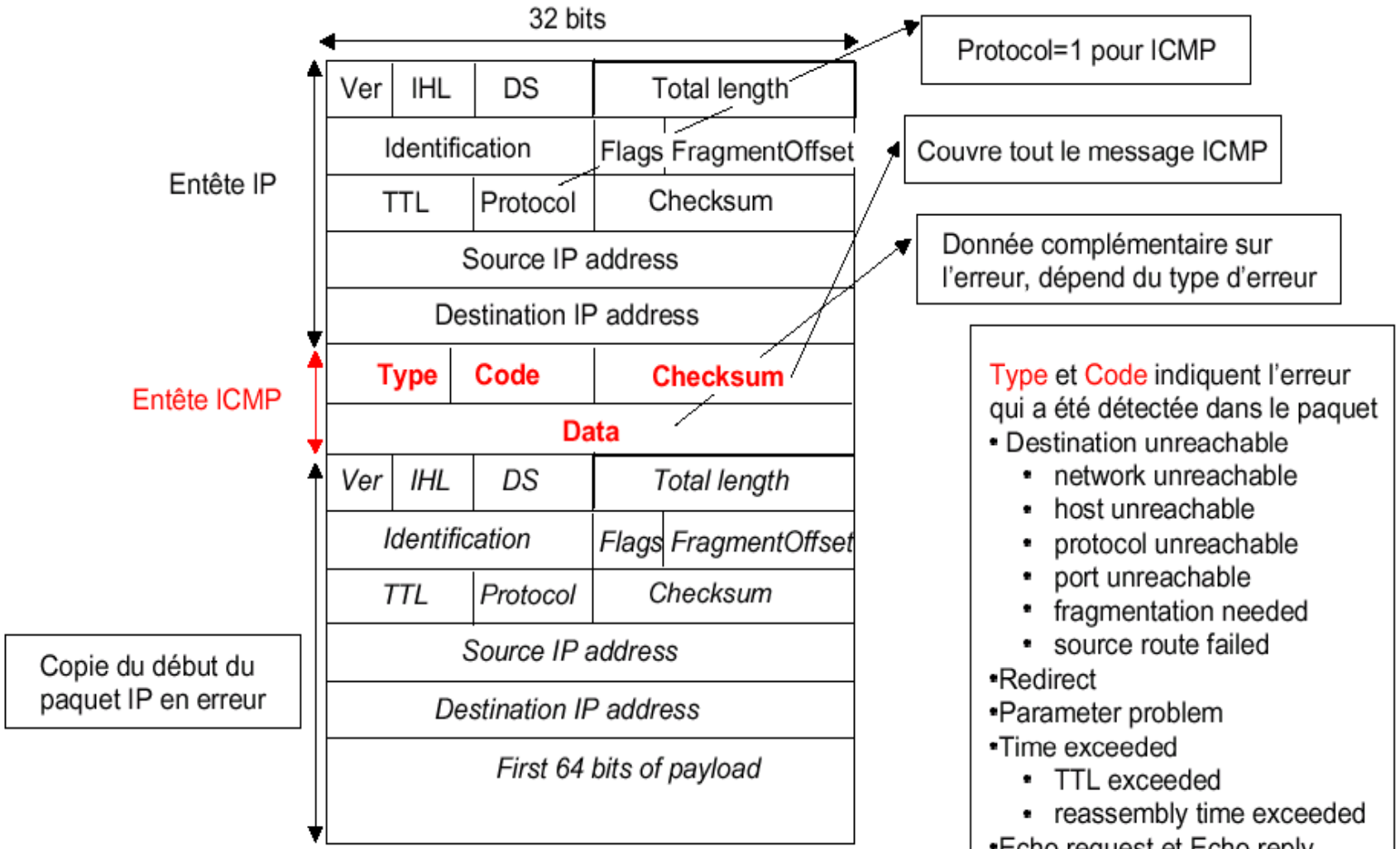
- ***Les processus impliqués dans la sélection du chemin pour chaque paquet sont les suivants:***
  - *Le routeur compare l'adresse IP du paquet reçu avec ses tables IP.*
  - *Il extrait l'adresse de destination du paquet.*
  - *Le masque de la première entrée dans la table de routage est appliqué à l'adresse de destination.*
  - *La destination masquée est comparée avec l'entrée de la table de routage.*
  - *Si une correspondance est établie, le paquet est transmis au port associé à cette entrée de table.*
  - *Si aucune correspondance n'est établie, l'entrée suivante de la table est examinée.*
  - *Si le paquet ne correspond à aucune des entrées de la table, le routeur recherche l'existence d'une route par défaut.*
  - *Si une route par défaut a été définie, le paquet est transmis au port qui lui est associé. La route par défaut est le chemin qui doit être utilisé lorsque aucune correspondance n'a pu être établie avec la table de routage. Elle est configurée par l'administrateur réseau.*
  - *Si aucun chemin par défaut n'existe, le paquet est éliminé. Un message est alors souvent envoyé à l'unité émettrice des données pour signaler que la destination n'a pu être atteinte.*

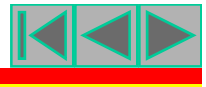


Principe :

- lorsqu'un routeur ne peut retransmettre un paquet vers sa destination, il doit informer la source du problème qu'il a rencontré
    - aussi utilisable par une station
- Problèmes détectables par un routeur :
    1. le routeur ne connaît pas de route vers la destination
    2. le format du paquet IP est incorrect
    3. la source aurait du utiliser un autre routeur intermédiaire pour joindre la destination
    4. le paquet reçu a un TTL=1
    5. le paquet reçu devrait être fragmenté, mais il contient le flag "Don't fragment«
- ICMP permet de rapporter ces problèmes à la source

ICMP : Format des messages





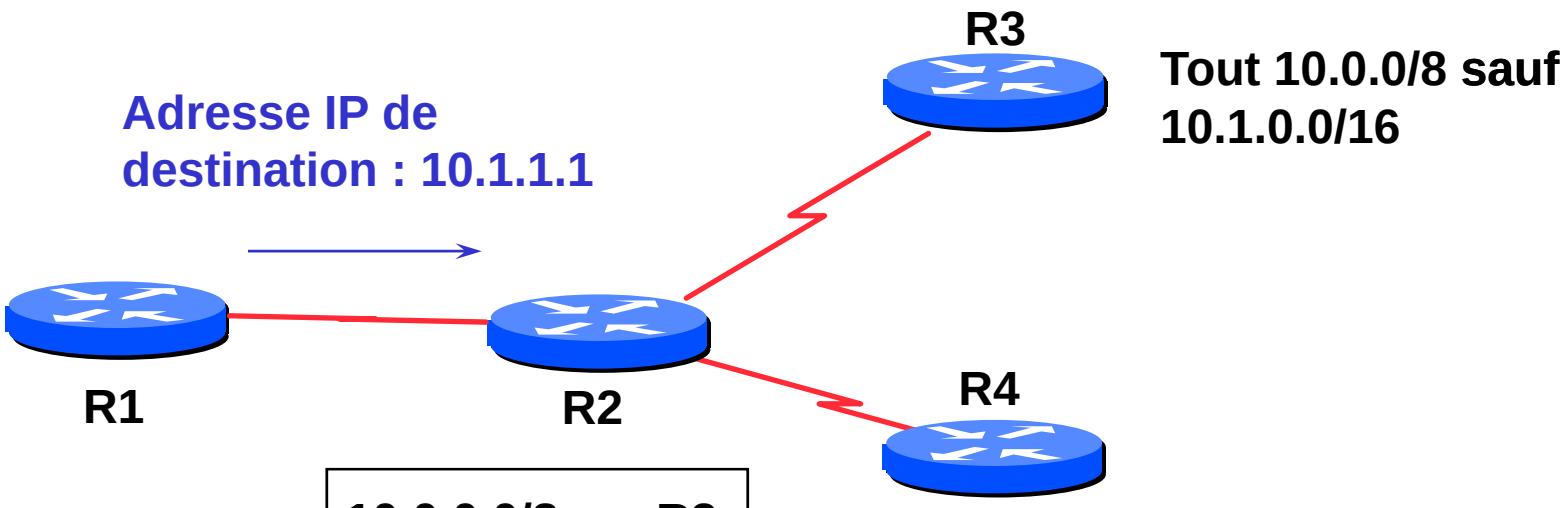
- Exemples d'utilisation des messages ICMP

1. destination unreachable :
   - le routeur qui envoie ce message n'a pas de route permettant de joindre la destination du paquet erroné
2. time exceeded :
   - le TTL du paquet erroné est arrivé à 0; utilisé par traceroute
3. Redirect :
   - pour atteindre la destination, il faut passer par un autre routeur dont l'adresse est donnée dans le msg ICMP

     echo request / echo reply : utilisé par ping
4. fragmentation impossible :
   - le paquet aurait du être fragmenté par le routeur mais son bit « Don't Fragment " avait la valeur "vrai"

Exemple d'utilisation de la table de routage "route spécifique"

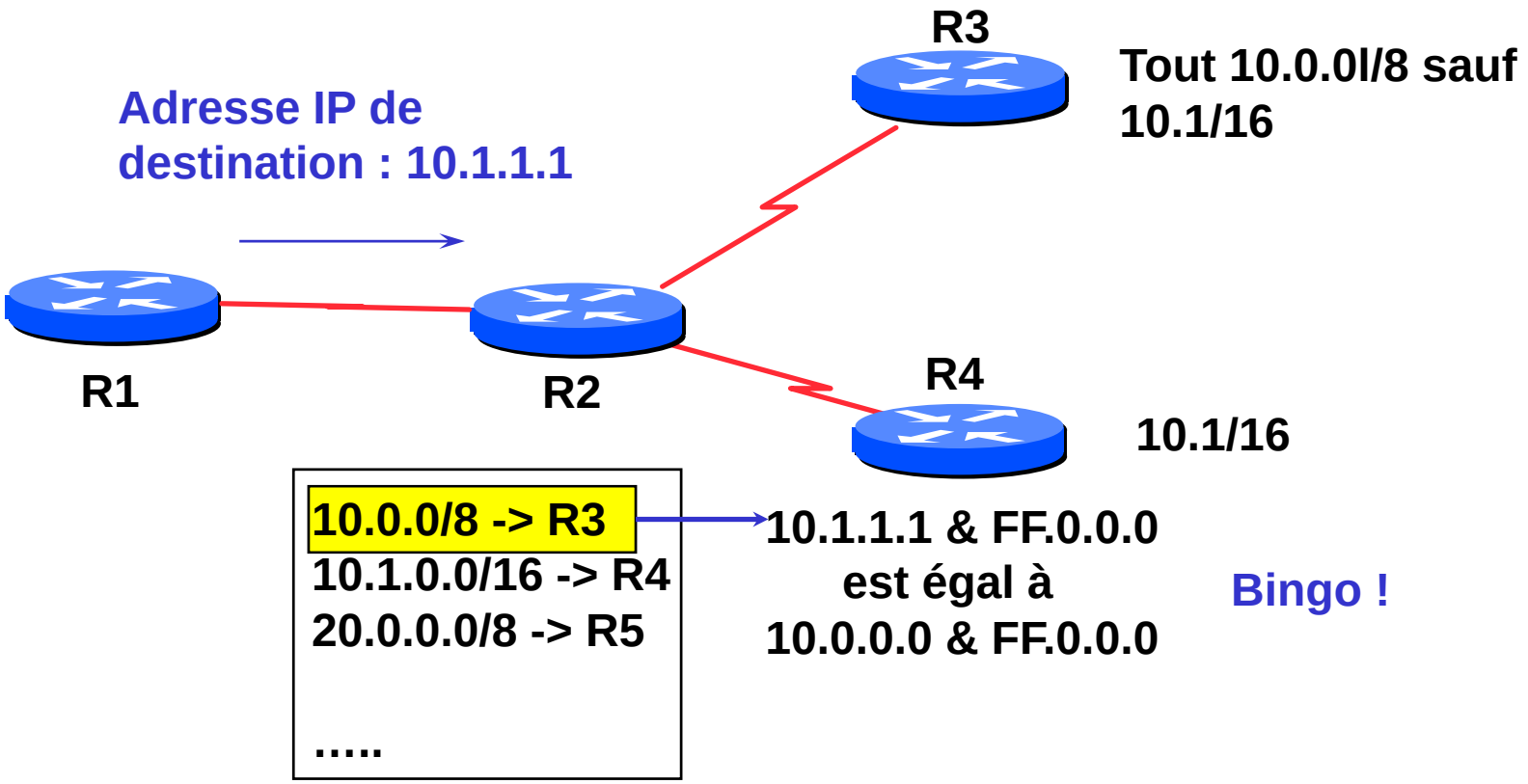
Priorité aux routes spécifiques



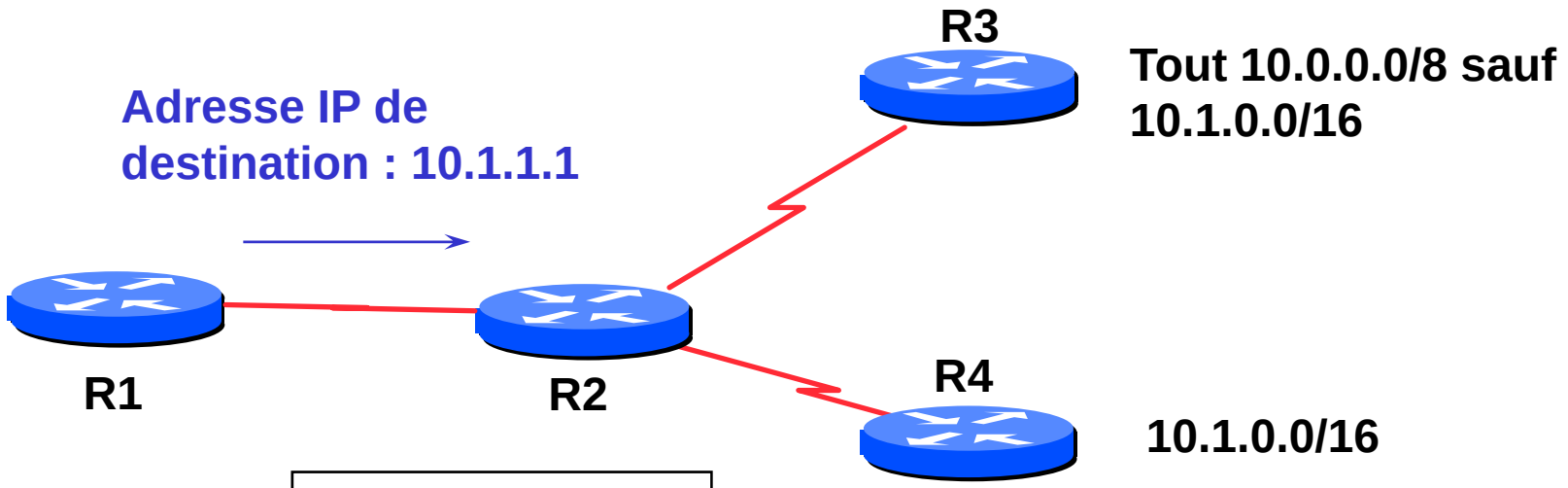
10.0.0.0/8	-> R3
10.1.0.0/16	-> R4
20.0.0.0/8	-> R5
30.0.0.0/8	-> R6
.....	

Table de routage IP de R2

Priorité aux routes spécifiques



Priorité aux routes spécifiques



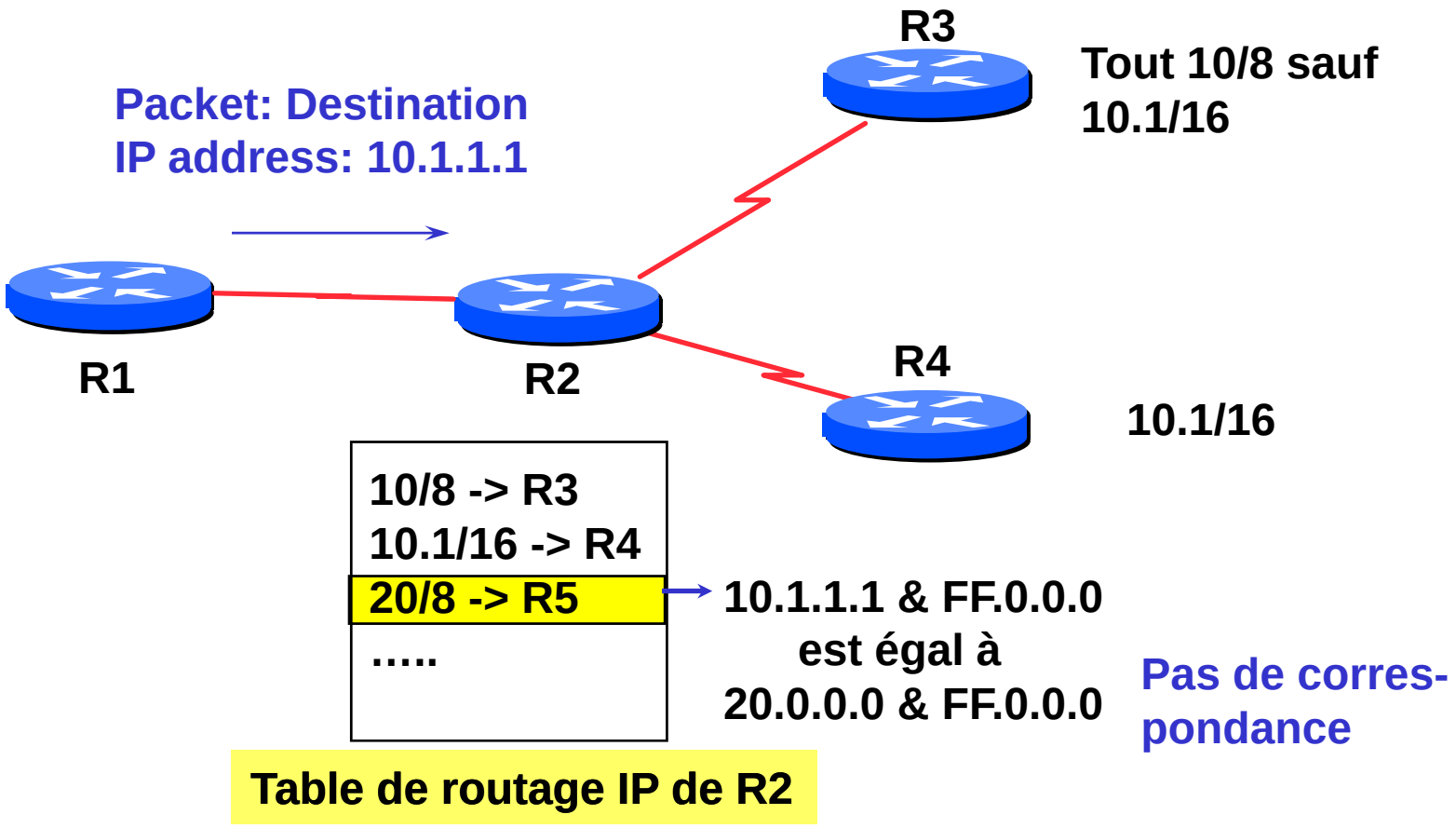
10.0.0.0/8 -> R3
10.1.0.0/16 -> R4
20.0.0.0/8 -> R5
…..

10.1.1.1 & FF.FF.0.0
est égal à
10.1.0.0 & FF.FF.0.0

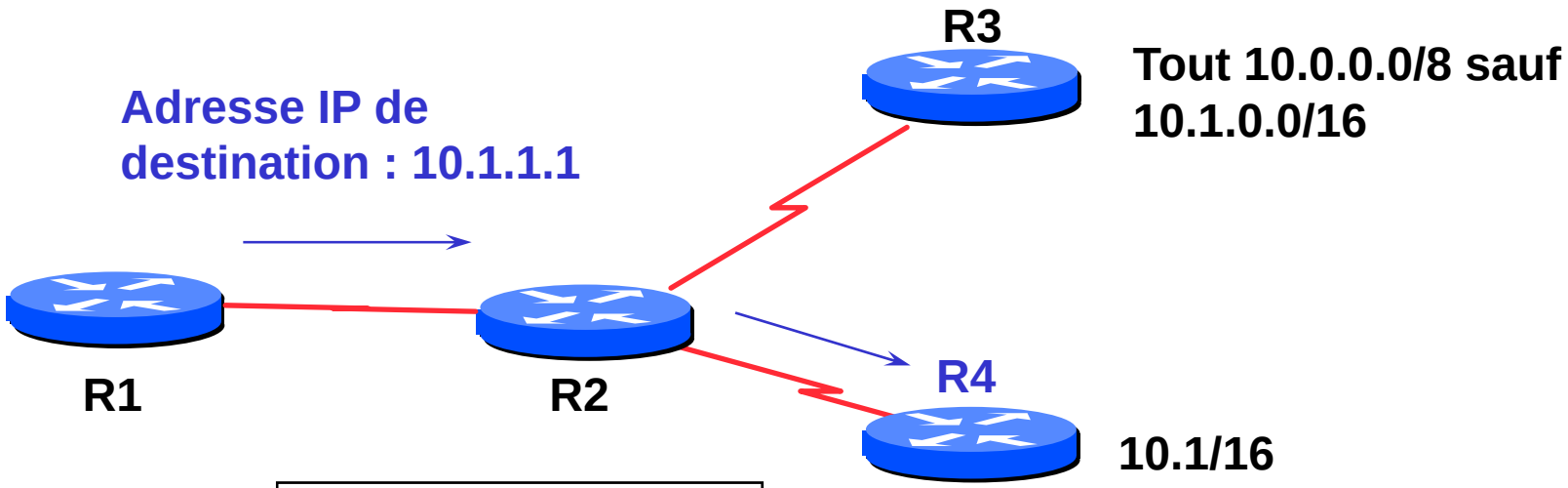
Re-Bingo !

Table de routage IP de R2

Priorité aux routes spécifiques



Priorité aux routes spécifiques

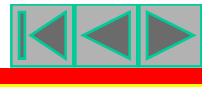


10.0.0.0/8 -> R3
10.1.0.0/16 -> R4
20.0.0.0/8 -> R5
.....

← Meilleure correspondance, masque réseau de 16 bits

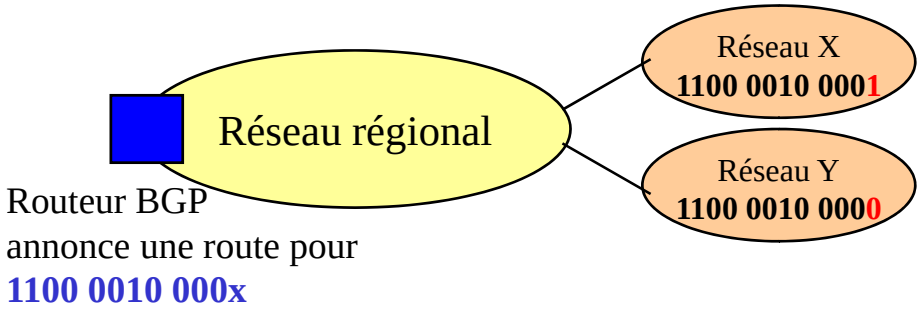
Table de routage IP de R2

annexe



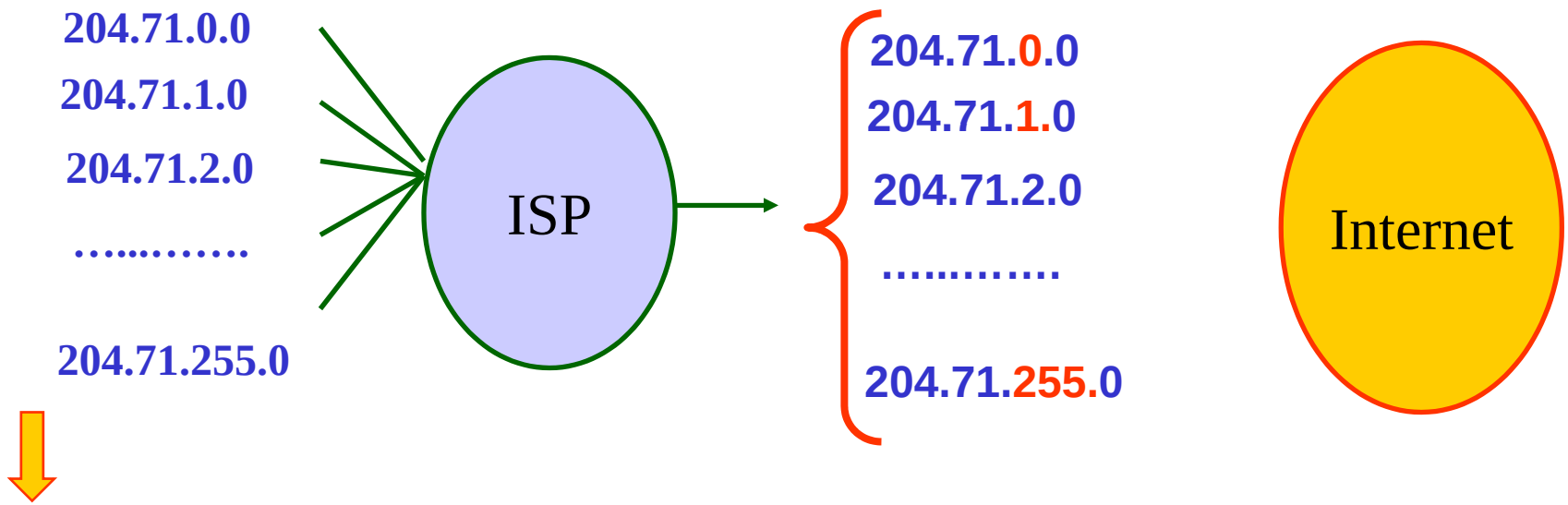
- CIDR :
  - Quand les adresses étaient assignées classe par classe, ily avait des pénuries d'adresse de classe B
  - Classless Inter-Domain Routing (CIDR)
    - **Aide à résoudre ce problème**
  - CIDR est un standard pour l'assemblage de « sur-réseaux »
    - **Permet d'agréger un ensemble de réseaux de classe C dans une seule entrée dans les tables de routage**
  - CIDR n'est pas un protocole en soi, CIDR est un moyen **pour « combiner » plusieurs** adresses de classe C
  - Les adresses combinées doivent avoir les mêmes **bits de poids fort**
    - **Doivent être contiguës :**
    - **Le groupe 194.128.64 – 67 peut être agrégé**
    - **Le groupe 194.128.66 – 69 ne le peut pas**
  - ***Les protcoles de routage doit utiliser un masque de 32 bits***
    - **RIP-2 (RIP n'accepte que les masques de taille fixe)**
    - **OSPF**

- **CIDR :**
  - **Implémenté dans la nouvelle version BGP-4**
    - **Définit des identificateurs de réseau de longueur variable**
    - **Compromis entre efficacité et complexité du routage**
  - **Agrégation de routes avec CIDR**
    - **Si un routeur utilise la même route pour plusieurs blocs d'adresses contigus, il peut annoncer une seule route**

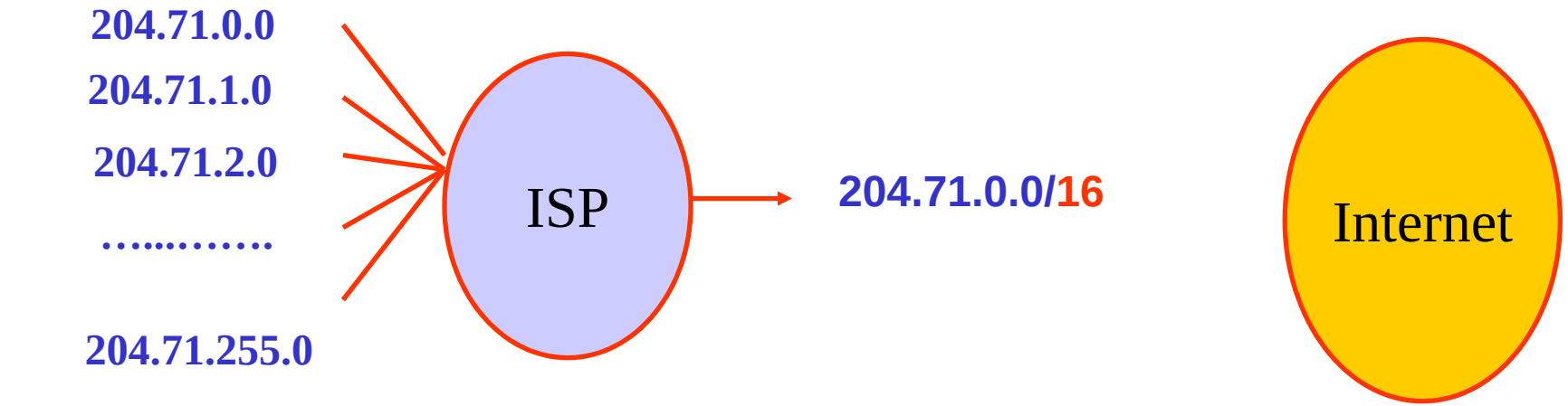


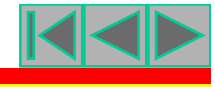
    - **Exemple**
      - **Adresses classe C : 194.0.0.0 – 195.255.255.255 --> Europe**
      - **Un routeur BGP américain considère les premiers 8 bit pour le routage**
      - **Un routeur BGP européen doit considérer des préfixes plus longs**

Routage sans CIDR



Routage avec CIDR

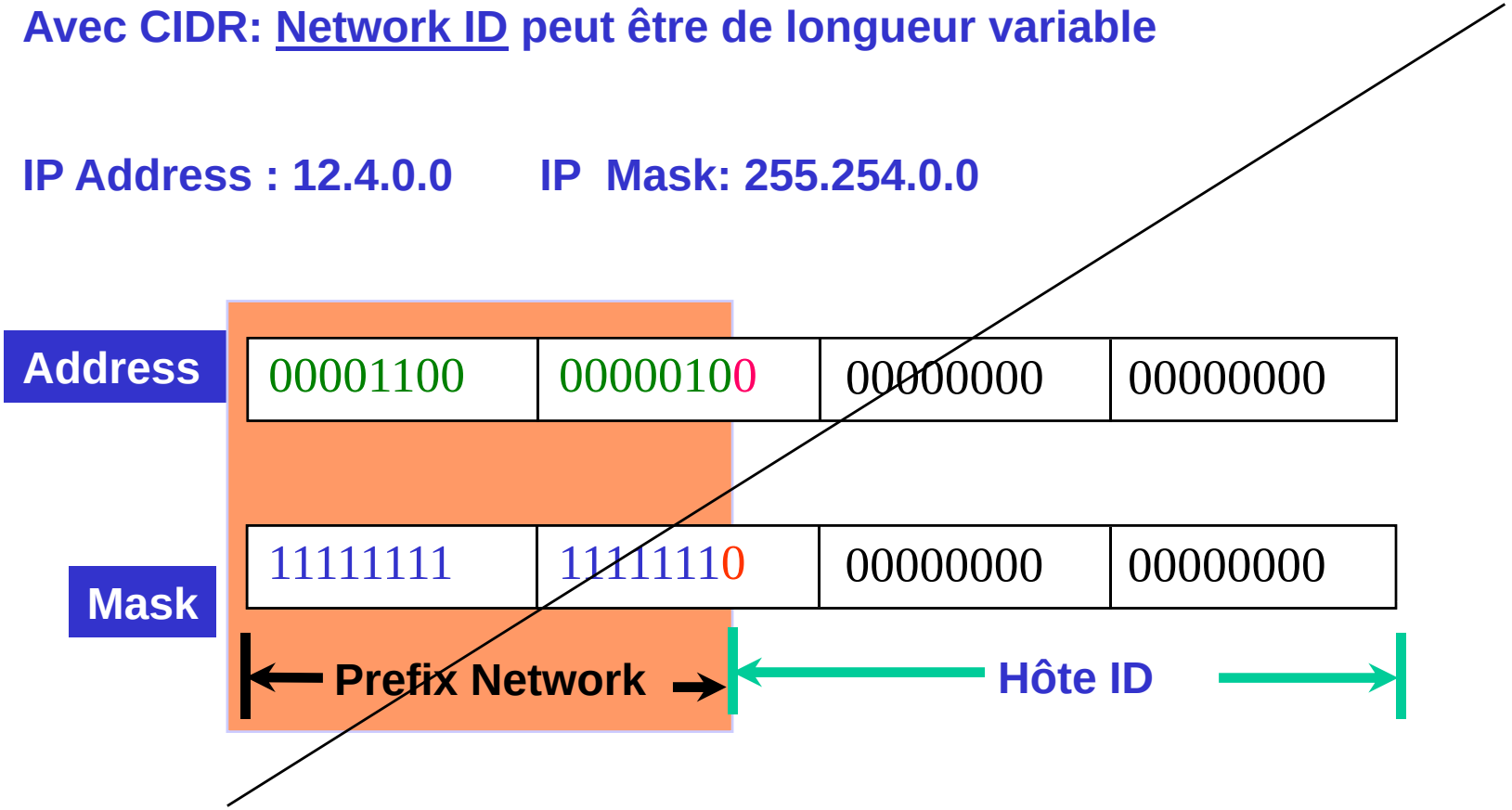




Sans-CIDR: Network ID est soit 8-, 16, 24- bits

Avec CIDR: Network ID peut être de longueur variable

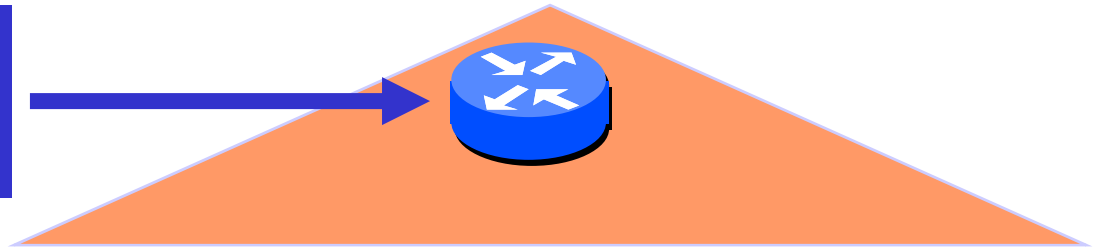
IP Address : 12.4.0.0 IP Mask: 255.254.0.0



S'écrit normalement : 12.4.0.0/15

-Longest Prefix Match (Classless) Forwarding -

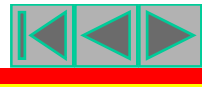
Destination =12.5.9.16
----
payload



	Prefix	Next Hop	Interface
OK	0.0.0.0/0	10.14.11.33	at0
Meilleur	12.0.0.0/8	10.14.22.19	at1
Encore meilleur	12.4.0.0/15	10.1.3.77	eth0
Le meilleur !	12.5.8.0/23	attached	S0

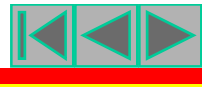
Table de forwarding

VLSM : Variable Length Subnet Mask



- **VLSM : Masque de Sous-Réseaux à Longueur Variable (RFC 1009)**
  - **Un réseau IP peut utiliser plusieurs masques différents pour l'ensemble des sous-réseaux d'un site.**
  - **Les VLSM économisent les adresses machines en partageant un réseau avec des masques de longueurs variables**
    - **Ils permettent de déployer des sous-réseaux de différentes tailles**
    - **Ils tiennent compte du nombre d'hôtes de chaque sous-réseau**
    - **Ils permettent d'ajuster la taille au cas ou il y aurait une extension**

- **Les protocoles de routages doivent supporter les VLSM**
  - **Utilisation des protocoles de protocoles standards (non propriétaires) RIP Version 2 ou OSPF**
- **Précautions :**
  - **Vérifier qu'il n'y a pas de chevauchement inter plage dans le découpage des sous réseau**



Block d'adresse : 128.20.224.0 / 20.

Réseaux :

2 contenant chacun 1000 noeuds;
2 contenant chacun 500 noeuds;
3 contenant chacun 250 noeuds
4 contenant chacun 50 noeuds.

Allocation des adresses ?

1000 nœuds ont besoin de 10 bits => 32 –10 =préfixe sur 22 bit est nécessaire

128.20.1110 00 00. 0000 0000/22 = 128.20.224.0/22
128.20.1110 01 00. 0000 0000/22 = 128.20.228.0/22

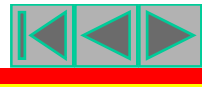
500 noeuds ont besoin de 9 bits => 32 –9 = préfixe sur 23 bit est nécessaire

128.20.1110100 0. 0000 0000/23 = 128.20.232.0/23
128.20.1110101 0. 0000 0000/23 = 128.20.234.0/23

250 nœud ont besoin 8 bits => 32 –8 = préfixe sur 24 bit est nécessaire

128.20.11101100. 0000 0000/24 = 128.20.236.0/24
128.20.11101101. 0000 0000/24 = 128.20.237.0/24
128.20.11101110. 0000 0000/24 = 128.20.238.0/24

50 nœuds ont besoin de 6 bits => 32 –6 = préfixe sur 26 bit est nécessaire

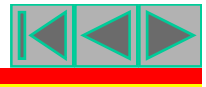


Annexe : Protocole ARP

ARP

- **RFC826 : Ethernet Address Resolution Protocol or converting network protocol addresses to 48bit Ethernet address for transmission on Ethernet hardware (11/1982).**
- **En résumé :**
- **Trouver une adresse MAC avec une adresse IP**
  - ***@ IP totalement indépendante de l'@ physique***
  - **ARP : Permet de trouver l'adresse physique (6 octets) d'une machine sur le même réseau en donnant son adresse IP (4 octets) uniquement.**

ARP : Cache



- Table de correspondance locale constituée après chaque réponse ARP :
  - @ physique <=> @ IP

- Stockage dans un cache :
  - Limite charge réseau, sinon 2 requêtes pour chaque datagramme IP
  - Augmente l'efficacité des communications locales
  - Cache remis à jour en fonction des besoins
    - Entrées valides durant un certain temps (4h : cisco)
    - En cas d'échec de la communication

ARP : Mécanismes

- Soit deux équipements sur le même segment Ethernet.
- Machine A veut envoyer un datagramme à la machine B.
  1. Elle connaît son adresse IP, mais pas son adresse Ethernet
  2. envoie une trame de broadcast (diffusion) Ethernet qui demande l'adresse Ethernet de B :
     - Adresse destinataire FF.FF.FF.FF.FF.FF avec Type = 0x0806
     - Indique l'adresse IP de B.
  3. Toutes les machines du réseau local reçoivent la requête.
  4. a) Seul B répond à A en lui donnant son adresse Ethernet.
  4. b) Si c'est une autre machine qui répond à la place de A on parle alors de "Proxy ARP".
     - Exemple : serveurs de terminaux et les stations connectées par accès distant

Message ARP

- Encapsulé dans trame ethernet

Longueur : 28 octets

(1) Ethernet = 1 (unique)

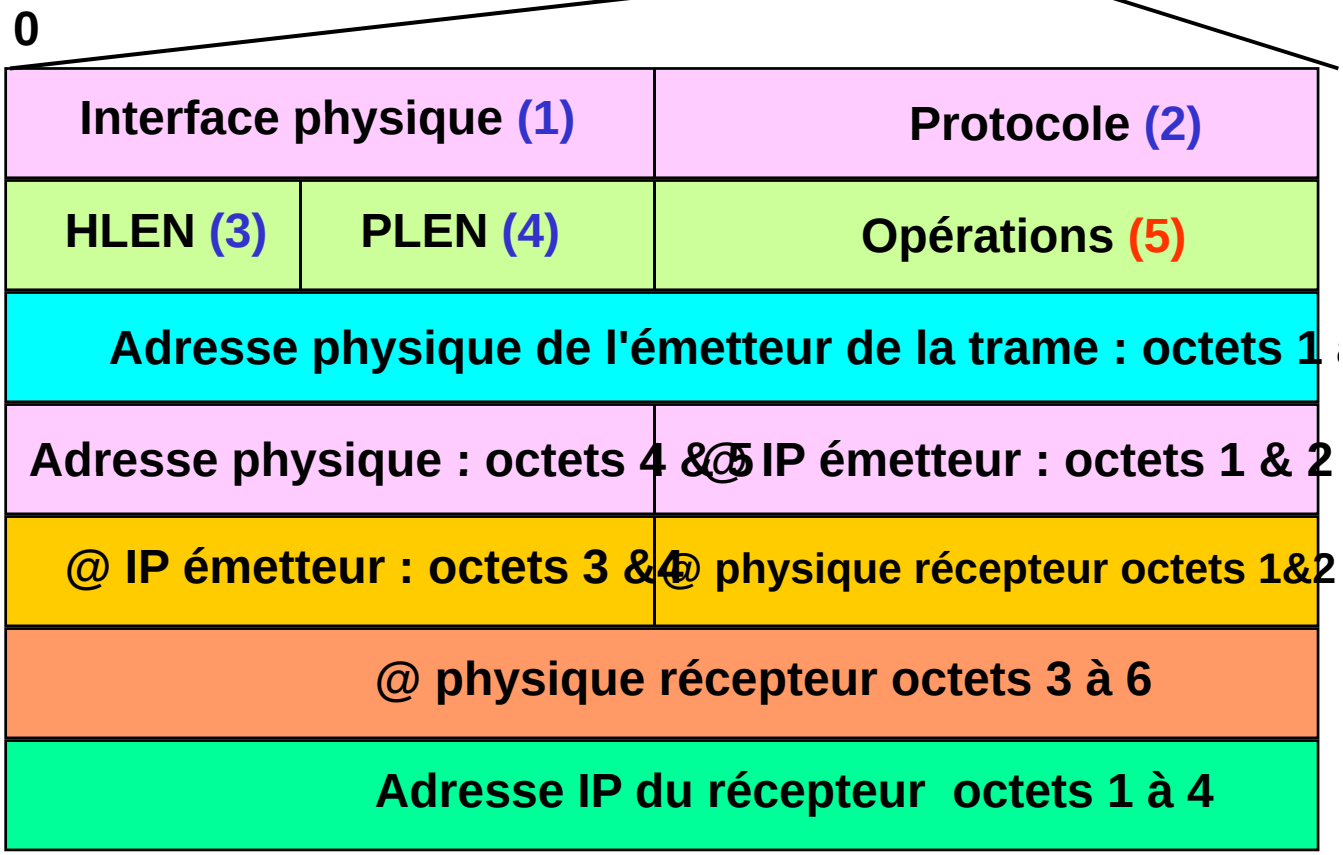
(2) IP = 0x0800

(3) HLEN : Longueur de l'@ physique

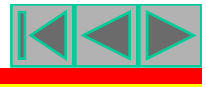
(4) PLEN : Longueur @ protocole

(5) Opérations :

- Requête ARP = 1
- Réponse ARP = 2

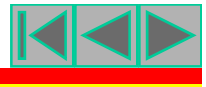


RARP (RFC903)



- RFC903 : Reverse Address Resolution Protocol (06/1984)
- Problème :
  - Déterminer un mécanisme permettant à la station d'obtenir son adresse IP depuis le réseau.
  - RARP est utilisé par des machines sans "mémoire secondaire"
  - L'adresse IP d'une machine est configurable par l'utilisateur
  - Est enregistrée généralement dans la "mémoire secondaire" (NVRAM, disque dur) où l'OS va la récupérer au démarrage.
- Trouver une adresse IP à partir de l'adresse Ethernet

RARP : Mécanismes



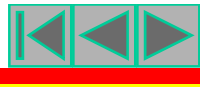
- Serveur RARP : fournit les adresses IP associées aux adresses physiques des stations du réseau (table).
- Utilisé au moment du "boot" par certains équipements.
    - Envoie son adresse MAC dans le champ « Adresse physique émetteur
- Type = 0x8035 dans la trame Ethernet
- Utilisé par
    - Stations sans disque ou anciens terminaux X
- Même format de message que ARP

- **Encapsulé dans trame ethernet**

Longueur : 28 octects

- **(1) Ethernet = 1 (unique)**
- **(2) IP = 0x0800**
- **(3) HLEN : Longueur de l'@ physique**
- **(4) PLEN : Longueur de l'@ protocole**
- **(5) Opérations:**
  - **Requête RARP = 3**
  - **Réponse RARP = 4**

0	
Interface physique (1)	**Protocole (2)**
HLEN (3) / PLEN (4)	Opérations (5)
Adresse physique de l'émetteur de la trame : octets 1 à	
Adresse physique : octets 4 & 5	**@ IP émetteur : octets 1 & 2**
@ IP émetteur : octets 3 &4	**@ physique récepteur octets 1&2**
Adresse physique récepteur octets 3 à 6	
Adresse IP du récepteur octets 1 à 4	



Fin annexe