# Routage intra-domaine et inter-domaine

Yvan Peter

IUT A - Université Lille 1

# Principes

- Deux modes de remise des paquets :
  - remise directe : les deux machines sont sur le même réseau
  - remise indirecte : il est nécessaire de passer par au moins un routeur pour délivrer le paquet à destination
- Comment choisir la route pertinente ?
- Comment apprendre de nouvelles routes ?

# La table de routage

- Les routeurs doivent connaître les destination existantes.
- Ces destinations sont stockées dans une **table de routage**
  - elle contient des informations relatives aux destinations et à la façon de les atteindre.
  - seuls les routeurs accessibles directement sont dans la table de routage (prochain saut).
- Le schéma d'adressage IP permet de ne conserver que des adresses de réseaux

# Les types de routage

## Routage statique

- Réalisé par un administrateur
- Pour définir des routes par défaut
- Pour forcer des routes vers des destinations particulières (test/debug)

## Routage dynamique

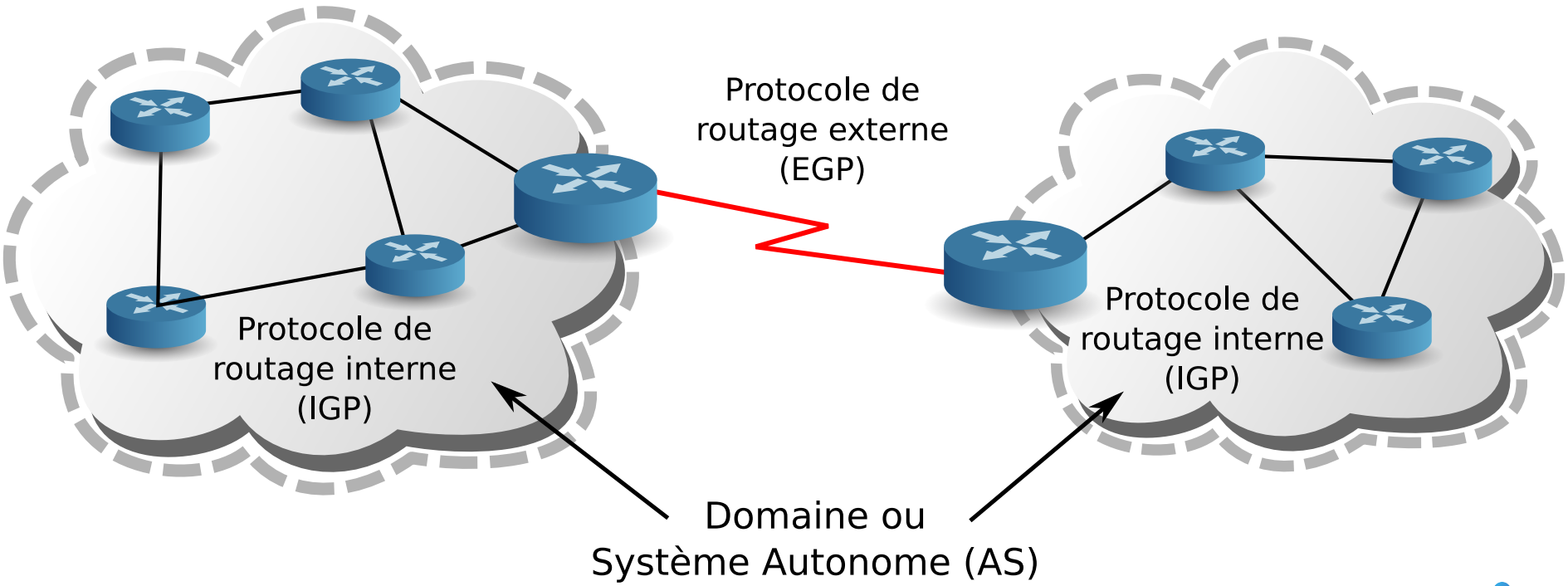
Réalisé par un protocole qui permet aux routeurs d'échanger de l'information pour construire leurs tables de routage

# La notion de domaine

- Un **domaine** ou **système autonome** correspond à un ensemble de réseaux connectés et qui partagent une politique de routage commune.
- Un système autonome est identifié par un numéro sur 32 bits (RFC 6793)
- Les numéros sont alloués par les Registres Internet

# Routage interne et externe

- On distingue les protocoles de routage **interne** et **externe**
  - Au niveau interne, on s'intéresse à des critères techniques (chemin le plus court)
  - Au niveau externe, on considère des critères administratifs (véhiculer les flux)



# Algorithmes de routage

On distingue deux types d'algorithmes de routage

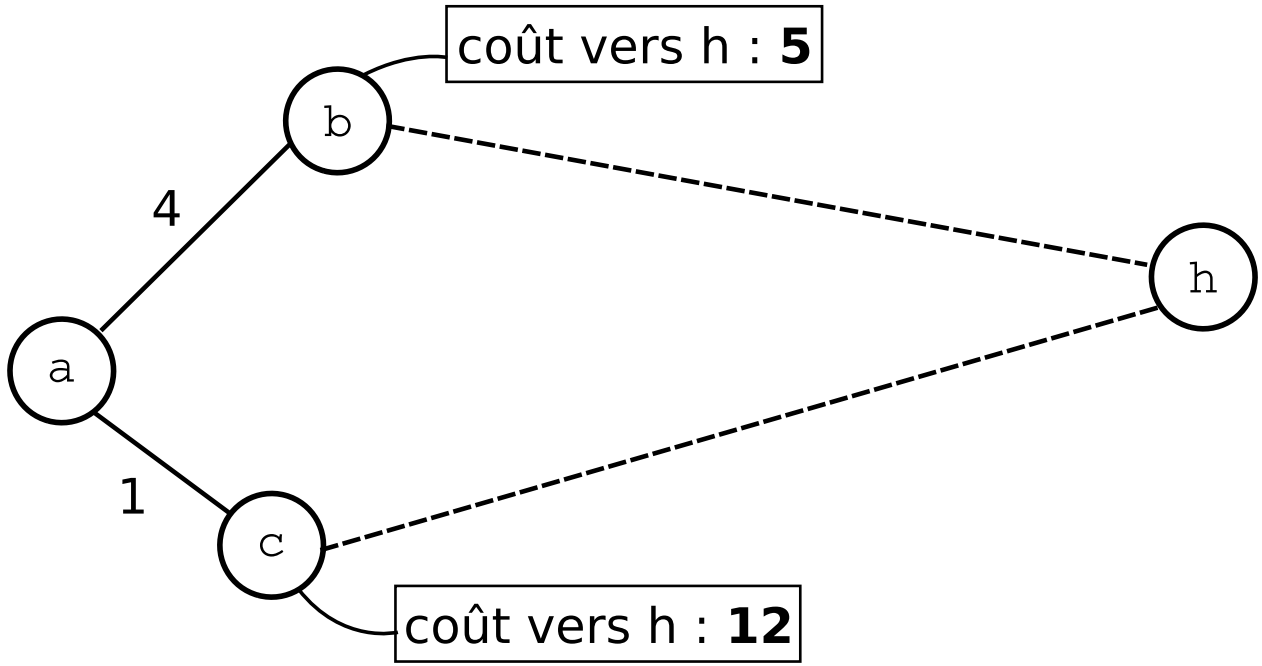
**les algorithmes dit à vecteur de distance**

basés sur les algorithmes de calcul du plus court chemin dans un graphe de Bellmann-Ford (1957) et Ford/Fulkerson (1962) pour la version distribuée

**les algorithmes dit à état de liaison**

basés sur l'algorithme de calcul du chemin le plus court d'abord (shortest path first) de Dijskstra (1959)

# Les algorithmes à vecteur de distance



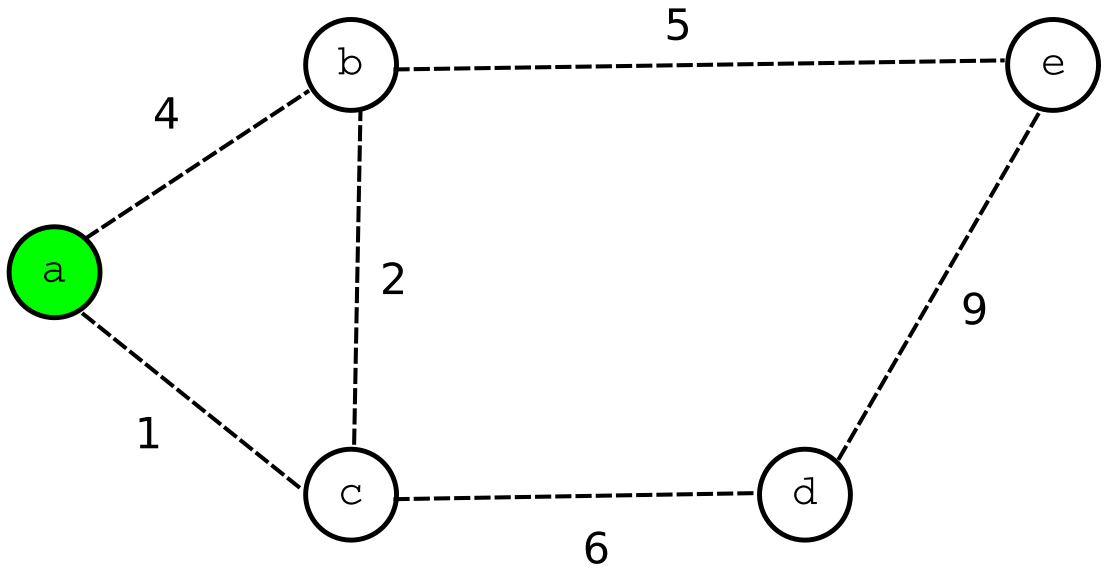

Coût du chemin du nœud ⓐ au nœud ⓗ :

- via ⓑ : 5 + 4 = 9
- via ⓒ : 12 + 1 = 13

# Les algorithmes à vecteur de distance

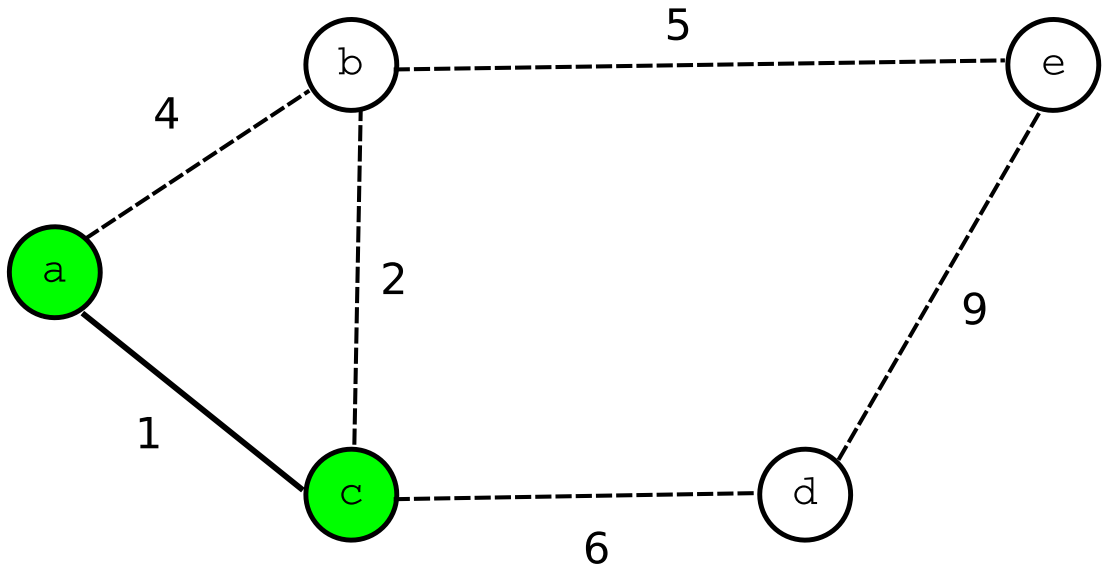
- Chaque nœud maintient une table de routage :
  - nœuds connus
  - nœud voisin sur le chemin le plus court
  - metrique totale pour atteindre le nœud
- Au départ chaque table ne contient que l'entrée qui correspond au nœud local
- Les tables sont mises à jour par des diffusions régulières d'informations entre les différents nœuds

# Algorithme à état de liaison



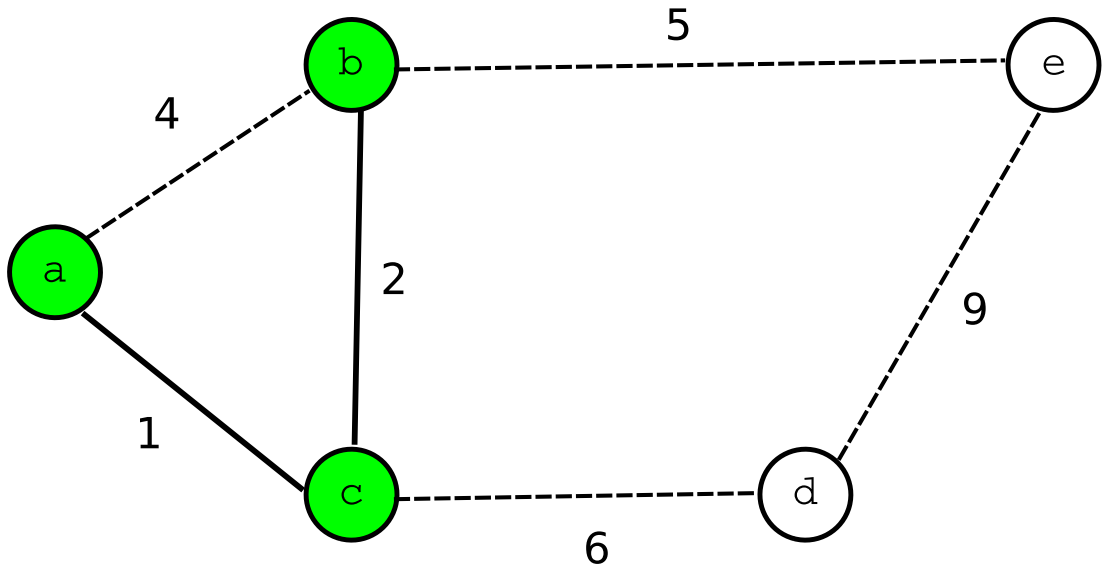

L'algorithme de Dijkstra contruit un arbre des chemins les plus courts depuis un point de départ (ici le nœud ⓐ) par itérations successives en sélectionnant à chaque fois le nœud le plus "proche" (au vu de la métrique).

# Algorithme à état de liaison



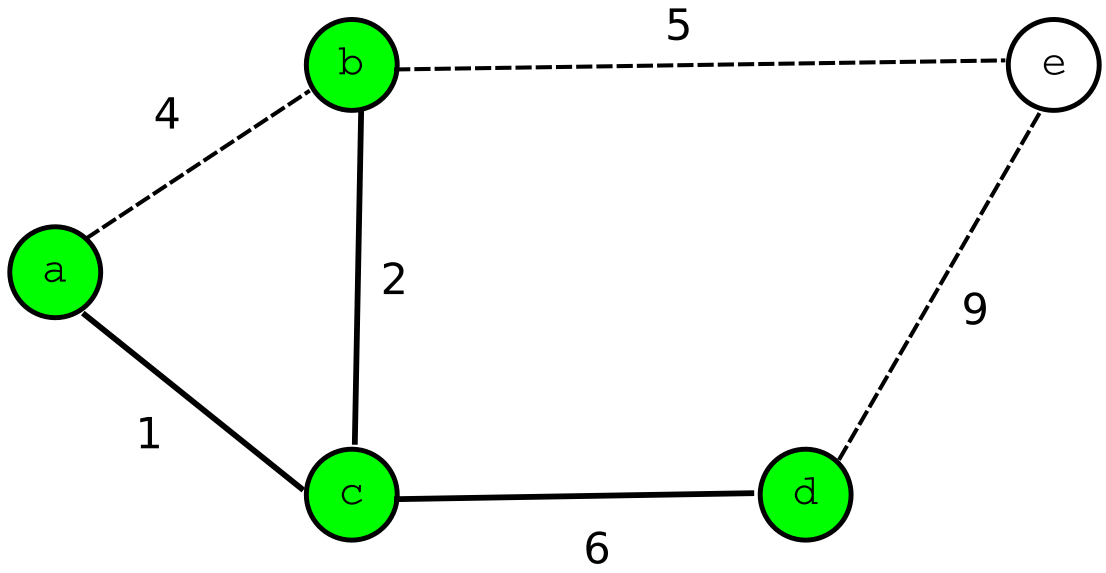

L'algorithme de Dijkstra contruit un arbre des chemins les plus courts depuis un point de départ (ici le nœud ⓐ) par itérations successives en sélectionnant à chaque fois le nœud le plus "proche" (au vu de la métrique).

# Algorithme à état de liaison



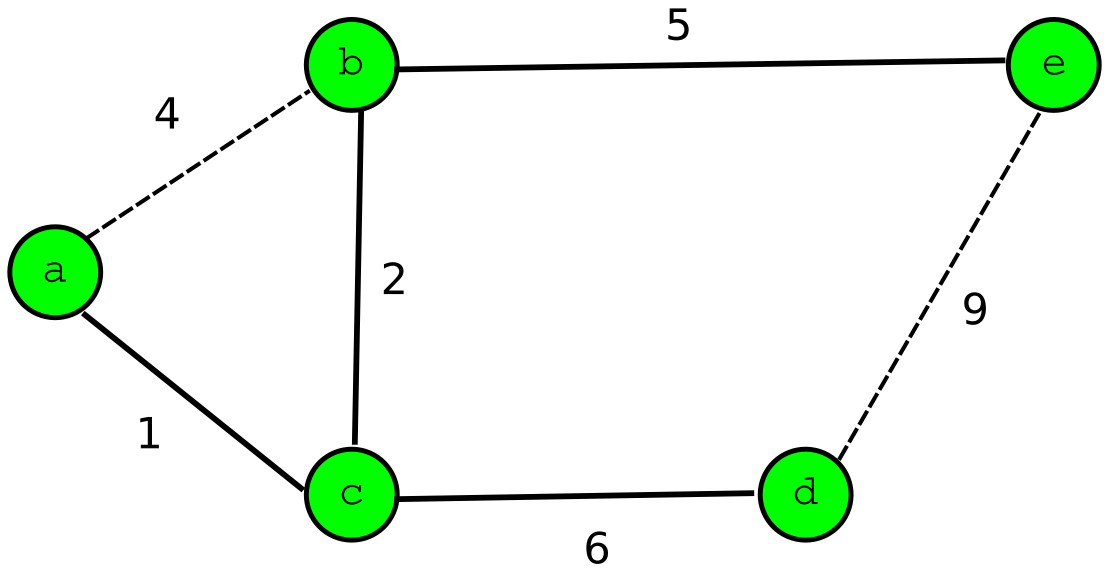

L'algorithme de Dijkstra contruit un arbre des chemins les plus courts depuis un point de départ (ici le nœud ⓐ) par itérations successives en sélectionnant à chaque fois le nœud le plus "proche" (au vu de la métrique).

# Algorithme à état de liaison




L'algorithme de Dijkstra contruit un arbre des chemins les plus courts depuis un point de départ (ici le nœud ⓐ) par itérations successives en sélectionnant à chaque fois le nœud le plus "proche" (au vu de la métrique).

# Algorithme à état de liaison



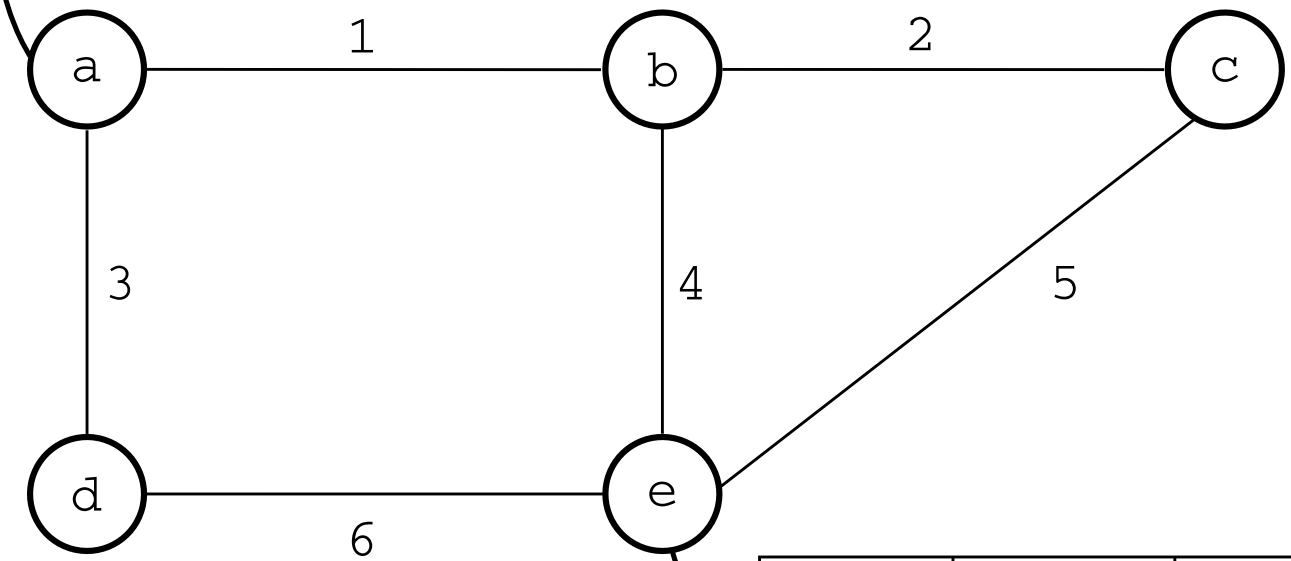

L'algorithme de Dijkstra contruit un arbre des chemins les plus courts depuis un point de départ (ici le nœud ⓐ) par itérations successives en sélectionnant à chaque fois le nœud le plus "proche" (au vu de la métrique).

# Algorithme à état de liaison

Pour mettre en œuvre l'algorithme de Dijkstra entre les routeurs, ceux-ci doivent se constituer une base de données topologique concernant les liaisons existantes (et leur coût) en échangeant avec leurs voisins.

# Construction des tables de routage

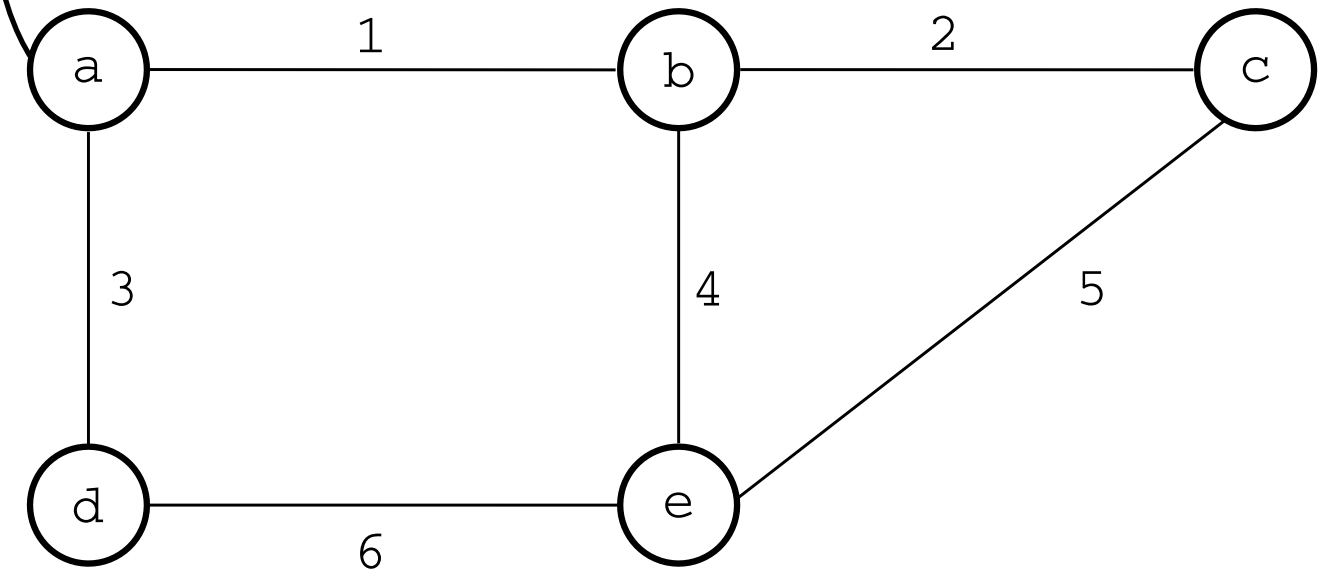
| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |




| de **e** à | liaison | coût |
|---|---|---|
| e | locale | 0 |

# Construction des tables de routage

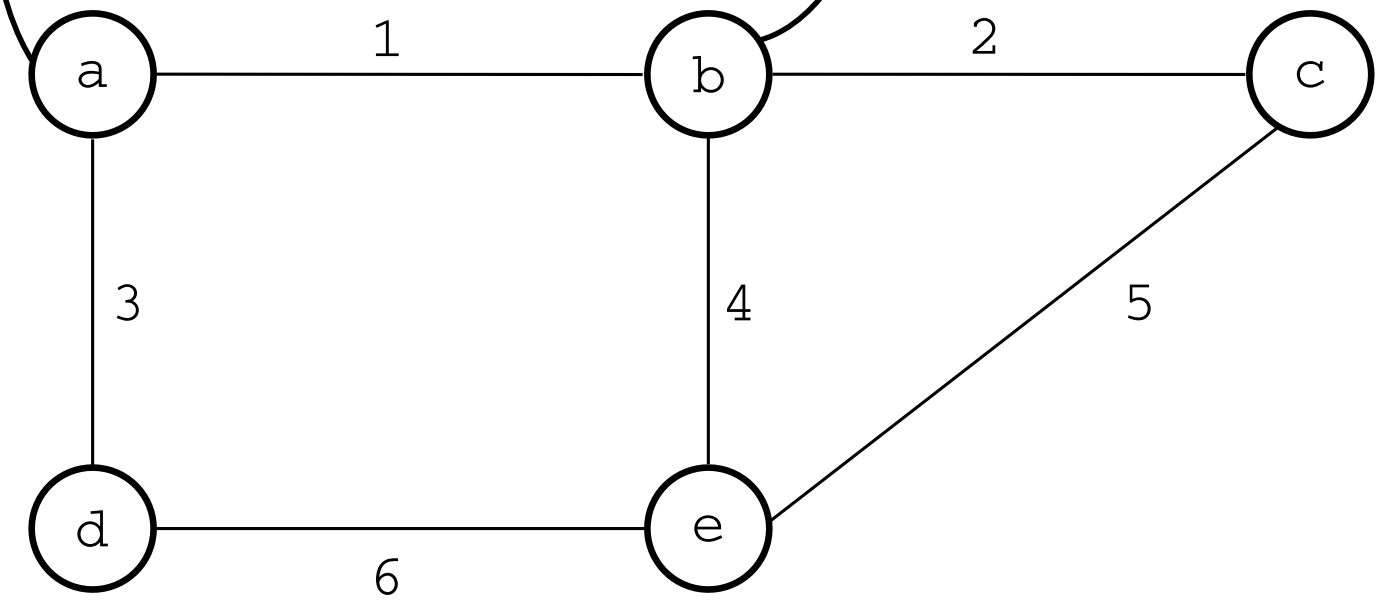
| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 1 | 1 |
| d | 3 | 1 |
| c | 1 | 2 |
| e | 1 | 2 |



# L'effet rebond

| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 1 | 1 |
| d | 3 | 1 |
| c | 1 | 2 |
| e | 1 | 2 |

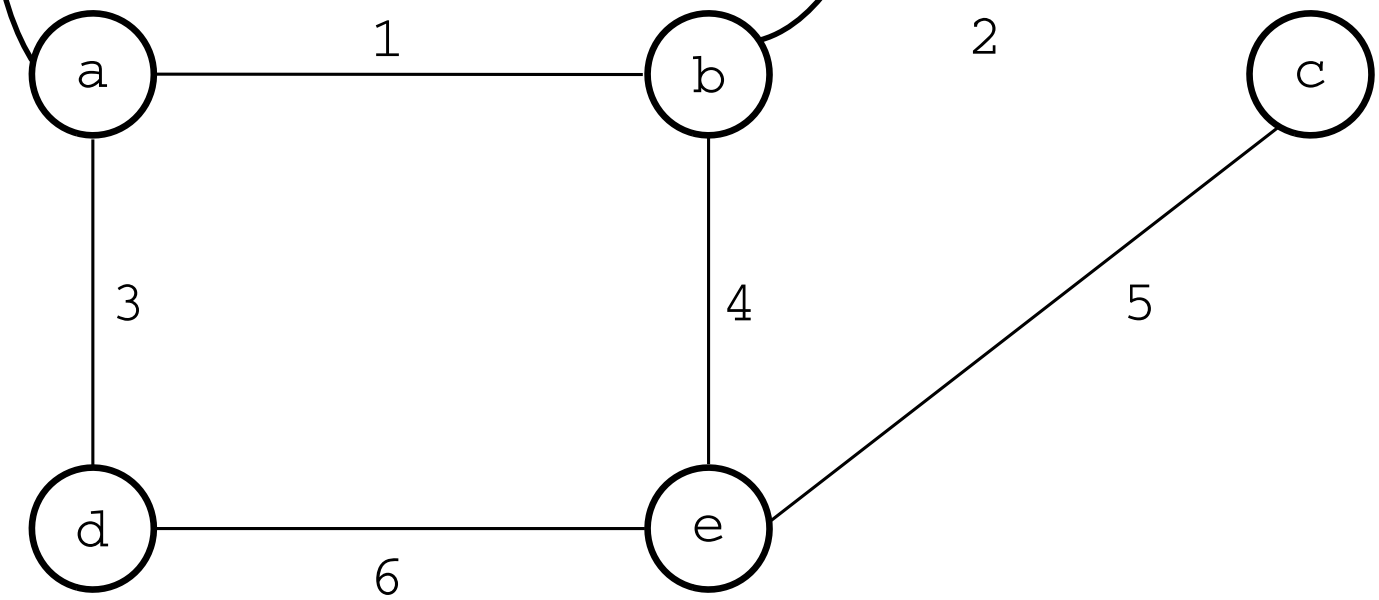
| de **b** à | liaison | coût |
|---|---|---|
| a | 1 | 1 |
| b | locale | 1 |
| d | 4 | 2 |
| c | 2 | 1 |
| e | 4 | 1 |



# L'effet rebond

| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 1 | 1 |
| d | 3 | 1 |
| c | 1 | 2 |
| e | 1 | 2 |

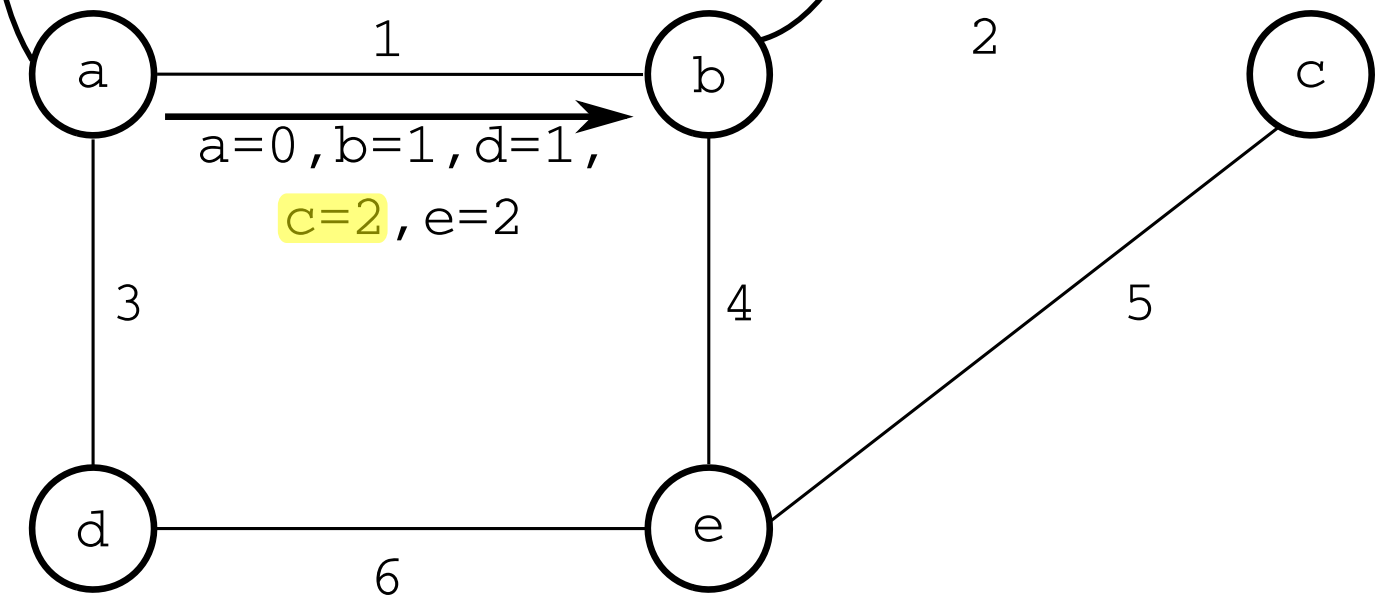
| de **b** à | liaison | coût |
|---|---|---|
| a | 1 | 1 |
| b | locale | 1 |
| d | 4 | 2 |
| c | 2 | ∞ |
| e | 4 | 1 |



# L'effet rebond

| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 1 | 1 |
| d | 3 | 1 |
| c | 1 | 2 |
| e | 1 | 2 |

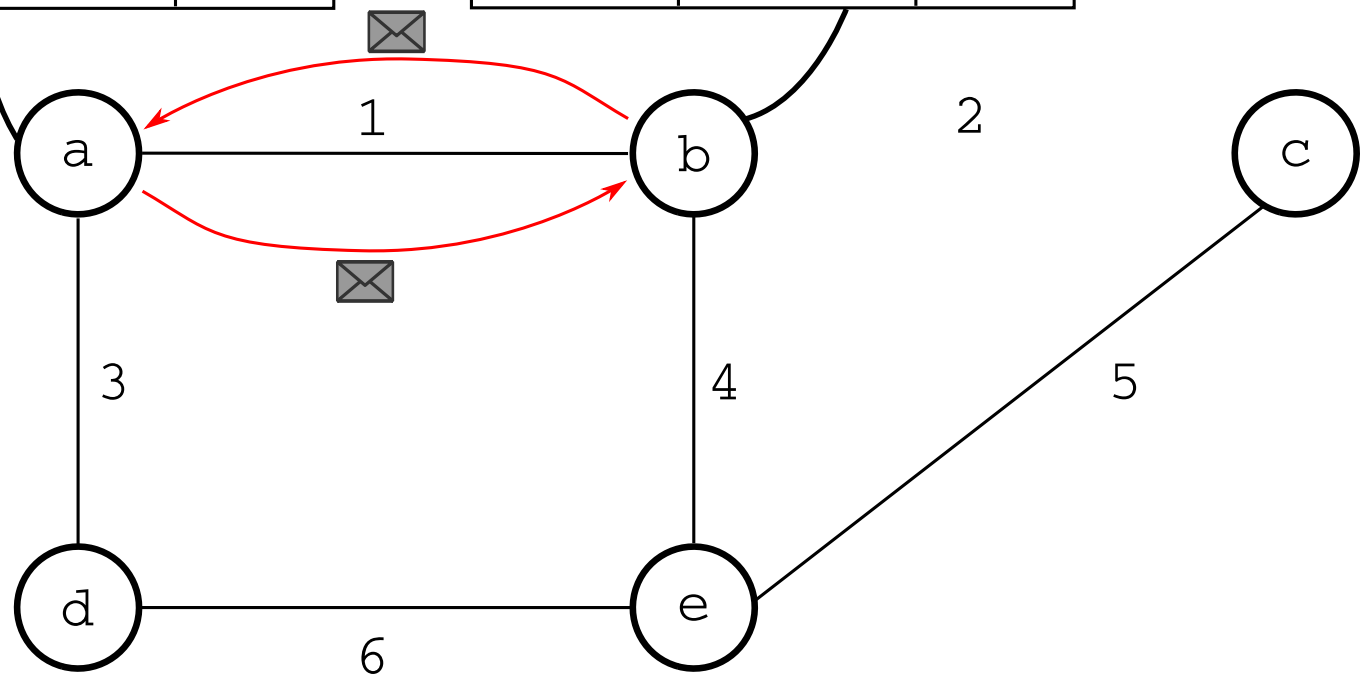
| de **b** à | liaison | coût |
|---|---|---|
| a | 1 | 1 |
| b | locale | 1 |
| d | 4 | 2 |
| c | 2 | ∞ |
| e | 4 | 1 |



# L'effet rebond

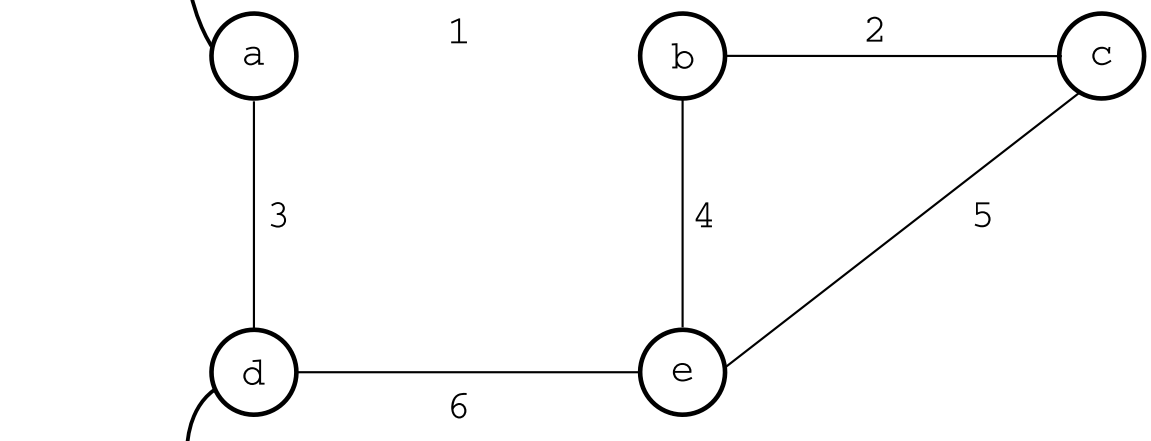
| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 1 | 1 |
| d | 3 | 1 |
| c | 1 | 2 |
| e | 1 | 2 |

| de **b** à | liaison | coût |
|---|---|---|
| a | 1 | 1 |
| b | locale | 1 |
| d | 4 | 2 |
| c | 1 | 3 |
| e | 4 | 1 |



# Le comptage à l'infini

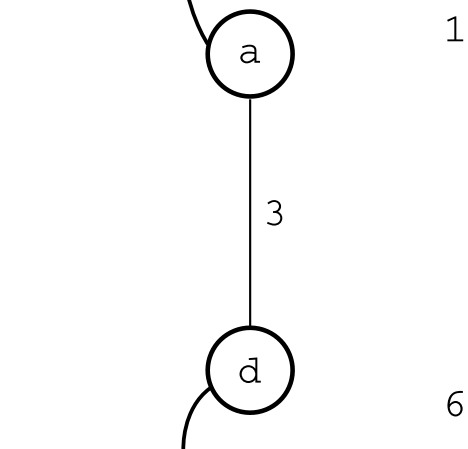
| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 3 | 3 |
| d | 3 | 1 |
| c | 3 | 3 |
| e | 3 | 2 |

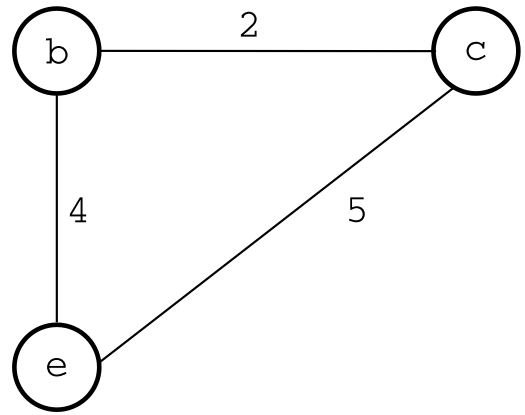



| de **d** à | liaison | coût |
|---|---|---|
| a | 3 | 1 |
| b | 6 | 2 |
| d | locale | 0 |
| c | 6 | 2 |
| e | 6 | 1 |

# Le comptage à l'infini

| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 3 | 3 |
| d | 3 | 1 |
| c | 3 | 3 |
| e | 3 | 2 |

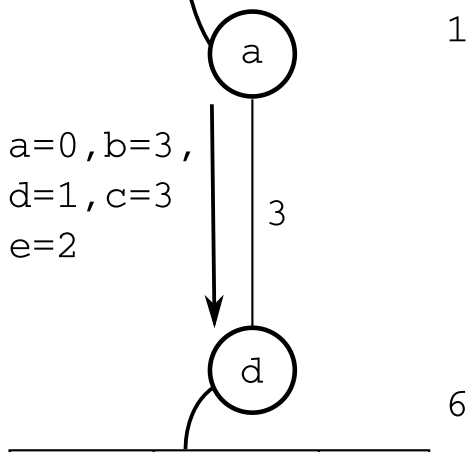


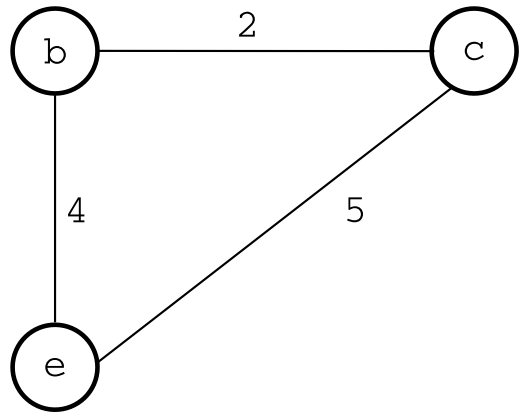




| de **d** à | liaison | coût |
|---|---|---|
| a | 3 | 1 |
| b | 6 | ∞ |
| d | locale | 0 |
| c | 6 | ∞ |
| e | 6 | ∞ |

# Le comptage à l'infini

| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 3 | 3 |
| d | 3 | 1 |
| c | 3 | 3 |
| e | 3 | 2 |

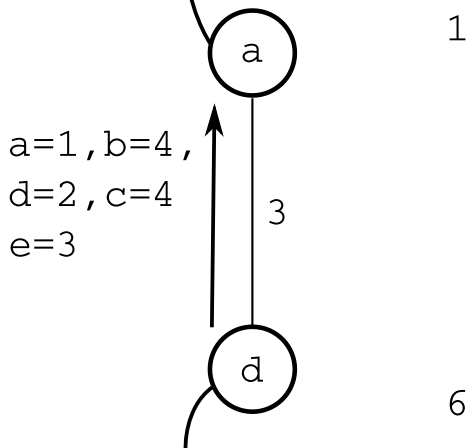


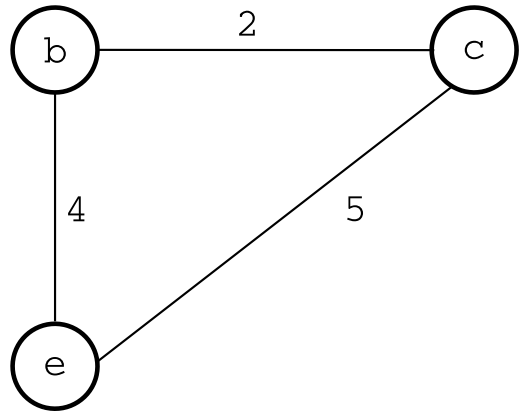

| de **d** à | liaison | coût |
|---|---|---|
| a | 3 | 1 |
| b | 3 | 4 |
| d | locale | 0 |
| c | 3 | 4 |
| e | 3 | 3 |

# Le comptage à l'infini

| de **a** à | liaison | coût |
|---|---|---|
| a | locale | 0 |
| b | 3 | 5 |
| d | 3 | 1 |
| c | 3 | 5 |
| e | 3 | 4 |




| de **d** à | liaison | coût |
|---|---|---|
| a | 3 | 1 |
| b | 3 | 4 |
| d | locale | 0 |
| c | 3 | 4 |
| e | 3 | 3 |



# Les correctifs

## L'horizon partagé

On ne donne pas d'informations à un nœud X sur les destinations qu'on atteint via X

- Soit on ne mentionne pas ces destinations
- Soit on les annonce avec une distance infinie (*poisoned route*)

## Les mises à jour déclenchées

En cas de défaillance d'un lien, on propage l'information le plus rapidement possible

# Bilan

- Avantages
  - simple à mettre en œuvre
  - messages simples à construire
  - peu de gestion pour les tables de routage
- Inconvénients
  - risque de boucles dans le réseau
  - la convergence peut être lente
  - inadapté aux ruptures de liaisons

# Routing Information Protocol

- la table de routage ne contient que les meilleures routes
- la distance est exprimée en nombre de sauts (routeurs traversés)
- le distance maximale est fixée à 15 (16 = $\infty$)

## RIPv1 (RFC1058)

- émission en brodcast sur le port 520 (UDP)
- *classful routing* (i.e., pas de VLSM)

## RIPv2 (RFC2453)

- émission en multicast (adresse 224.0.0.9) sur le port 520 (UDP)
- transmet adresse ET masque (*classless routing*)
- ajout d'une authentification (en clair...)

# Routing Information Protocol : Messages

## Requête : demande de route à un voisin

- au démarrage du nœud : pour la route 0.0.0.0 et une métrique infinie
- pour des routes spécifiques (test)

## Réponse : annonce de route

- toutes les 30 secondes (sauf mise à jour déclenchée)
- un délai aléatoire supplémentaire (1 à 5 secondes) évite les synchronisations

# Protocoles à état de liaison

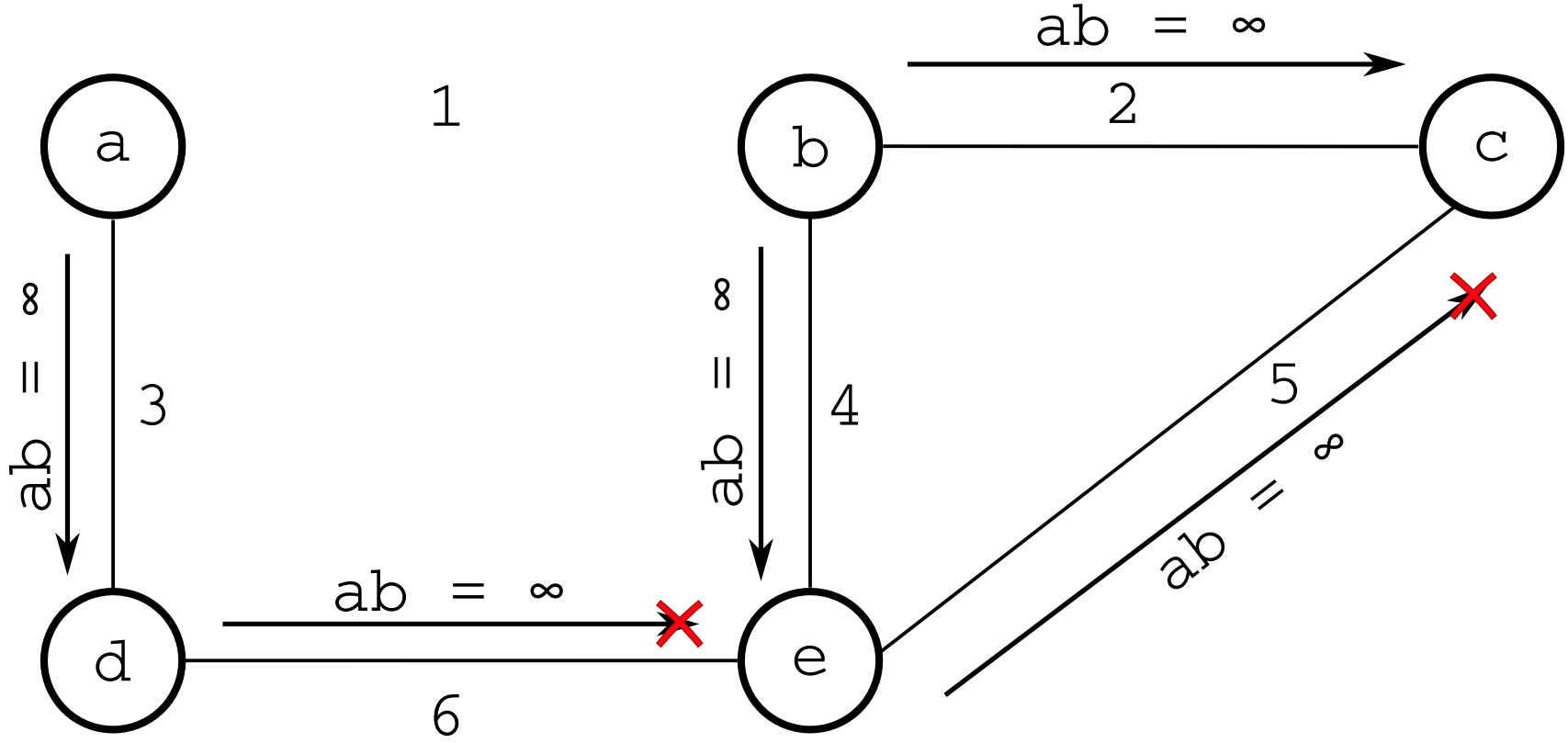
## Etat de liaison ?

- une *liaison* correspond à une interface du routeur
- l'*état* de la liaison décrit l'interface (@IP, masque, type de réseau...) et ses relations avec les routeurs voisins
- l'ensemble de ces états de liaison constitue une base de données

# Protocoles à état de liaison

- Tous les nœuds possèdent la même base de donnée topologique :
  - les liaisons ainsi que les nœuds auxquels ils sont rattachés
  - le coût de chaque liaison
  - l'état de chaque liaison (*up* ou *down*)
- Chaque nœud diffuse les liaisons et leurs états par inondation pour constituer cette base de donnée et pour la maintenir à jour
- En appliquant l'algorithme shortest path first, un nœud construit un arbre dont il est la racine vers toutes les destinations possibles.
- les chemins les plus courts sont utilisés pour constituer la table de routage.

# Protocoles à état de liaison




## Protocole d'inondation

Il est impératif de maintenir une vision cohérente de la topologie.
En cas de changement, on répercute sur les voisins.
Un numéro d'enregistrement permet de reconnaître une nouvelle mise à jour.

# Open Shortest Path First

- créé par l'IETF pour palier aux défaut de RIP
  - OSPFv1 (1989, RFC 1131)
  - OSPFv2 (1991, RFC 1247 → 1998, RFC 2328)
  - OSPFv3, prise en charge d'IPv6 (1999, RFC 2740 → 2008, RFC 5340)
- basé sur un mécanisme d'inondation pour la diffusion des états de liaisons pour maintenir un vision globale de la topologie
- utilise l'algorithme SPF pour calculer les routes
- les chemins de taille équivalente sont conservés et le trafic est réparti sur les différents liens

# OSPF : caractéristiques

- OSPF prend en compte différents types de réseaux
  - liaison point à point
  - réseaux à diffusion (*broadcast network*) (e.g., Ethernet)
  - réseaux multi-accès sans diffusion (*non-broadcast multiaccess* ou *NBMA*) (e.g. ATM ou Frame Relay)
  - réseaux point à multipoint (NBMA)
  - liaisons virtuelles
- permet de gérer au mieux la base de données topologique et les échanges de messages
- sur certains réseaux, on a un *routeur désigné* (*Designated Router* ou DR) qui gère les échanges

# OSPF : caractéristiques

## Coût d'une liaison

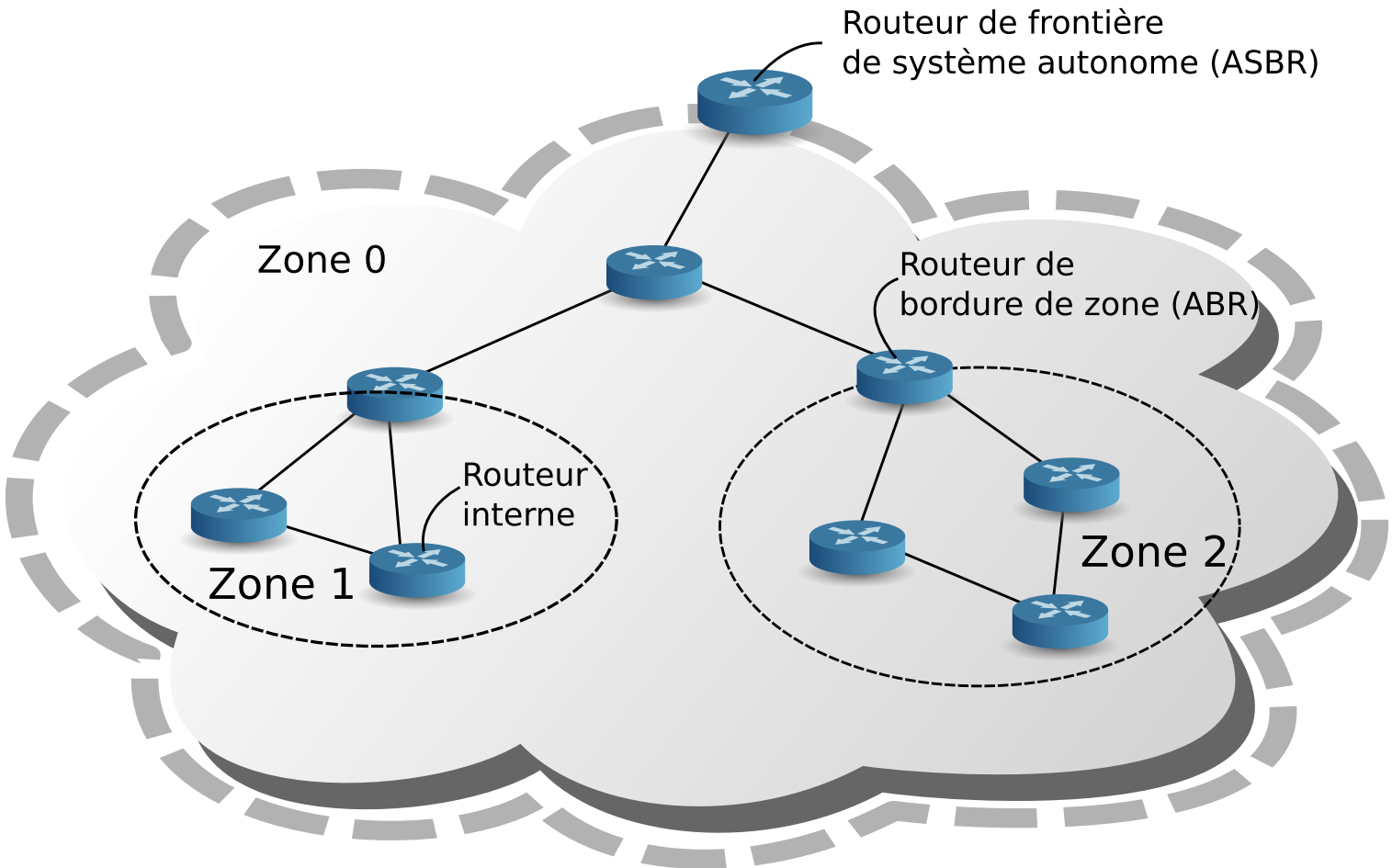
Le coût d'une liaison dépend de son débit

$$\text{coût} = \frac{1000000000}{\text{bande passante en bit/s}}$$

## Découpage en zones

OSPF permet de diviser le réseau en zones pour réduire la taille de la base de données topologique et l'étendue des phases d'inondation

# OSPF : découpage en zones




ASBR : Autonomous System Border Router
ABR : Area Border Router

# OSPF : types d'enregistrements

Du fait du découpage en zones et des différents types de réseaux, on a différents types d'enregistrements (LSA ou *Link State Advertisement*) dans la base de donnée topologique

- *Router LSA* généré pour chaque interface d'un routeur sur une liaison point à point. Annonce restreinte à la zone
- *Network LSA* pour les réseaux multi-accès. Annonce générée par le DR qui contient tous les routeurs et le DR. Annonce restreinte à la zone
- *Network Summary LSA* généré par les routeurs de bordure de zone (ABR). Permet d'annoncer les destinations à l'extérieur de la zone
- *ASBR Summary LSA* généré par les routeurs de bordure de zone (ABR). Annonce les routeurs de frontière de système autonome
- *AS External LSA* généré par les routeurs de frontière de système autonome pour indiquer les destinations extérieurs au système autonome
- ...

# OSPF : fonctionnement

## Au niveau du routeur

Le routeur maintient trois tables :

- Une table des voisins (avec qui il échange des informations)
- Une base de donnée topologique (état des liaisons)
- La table de routage

## En terme de protocole

- des messages périodiques pour vérifier la connectivité (hello)
- l'envoi de mises à jour uniquement en cas de changement de topologie

# OSPF : protocole de Hello

Le protocole de Hello est un échange de messages entre routeurs du même segment de réseau.

## Vérification de la connectivité

Il permet de définir :

- l'intervalle de Hello (temps entre deux échanges)
- l'intervalle de mort (temps au bout duquel la connectivité est considérée perdue)

## Choix du routeur désigné et du backup

Il permet également de choisir le routeur désigné (et son backup). Le routeur désigné centralise les échanges d'informations et les retransmet aux autres routeurs.

# OSPF : fonctionnement

## voisinage

Les routeurs sur le même segment deviennent voisins si

- ils ont le même **identifiant de zone**
- ils ont le même **mot de passe** pour l'authentification de zone (optionel)
- ils sont d'accord sur les **intervalles de Hello et de mort**

## adjacence

Après établissement du voisinage, les routeurs peuvent devenir adjacents (avec le DR). A partir de là, ils échangent leurs base de donnée d'état de liaisons.

# OSPF : fonctionnement