

# Recherche de chemin de longueur optimale

Dijkstra

# recherche du plus court chemin(1)

- La recherche du plus court chemin est utilisée dans plusieurs domaines et applications, dans notre contexte du cours, on va focaliser sur l'utilisation des algorithmes de Moore-Dijkstra dans les protocoles de routage.

# recherche du plus court chemin (2)

- L'algorithme de routage doit trouver le « meilleur » chemin pour aller d'un point source A à une destination B. Il existent plusieurs critères pour définir la notion de meilleur chemin dans un réseau.
- Le moins cher (argent),
- Le plus rapide (en temps),
- Le plus grand débit,
- Le plus sûr,
- Passer par le moins de nœuds possible,
- Le plus court (en distance).

# Algorithme de Moore-Dijkstra (1959)

- L'idée de l'algorithme est simple : on ne peut atteindre une destination qu'en passant par l'une des voies qui y conduit. On va donc partir du point de départ et aller vers tous les voisins en calculant le coût du trajet. Puis, depuis chacun de ces points, on calcule le coût minimum pour aller vers les voisins immédiats (nœuds dotés d'une voie directe). Le résultat est une **arborescence**, c'est-à-dire un arbre avec un sommet particulier appelé **racine**.

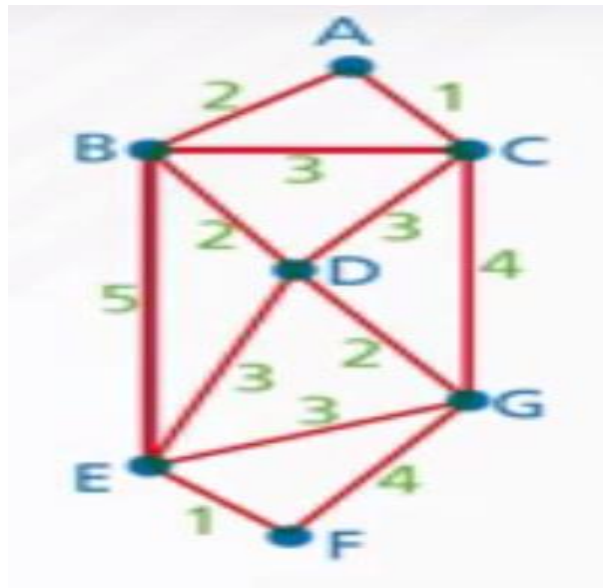
# Algorithme de Moore-Dijkstra (2)

Entrées :  $G = (S, A)$  un graphe avec une valuation positive  $v$  des arcs,  $s_{deb}$  un sommet de  $S$

```
 $P := \emptyset$   
 $d[a] := +\infty$  pour chaque sommet  $a$   
 $d(s_{deb}) = 0$   
Tant qu'il existe un sommet hors de  $P$   
  Choisir un sommet  $a$  hors de  $P$  de plus petite distance  $d[a]$   
  Mettre  $a$  dans  $P$   
  Pour chaque sommet  $b$  hors de  $P$  voisin de  $a$   
     $d[b] = \min(d[b], d[a] + poids(a, b))$   
  Fin Pour  
Fin Tant Que
```

# Algorithme de Moore-Dijkstra (3)

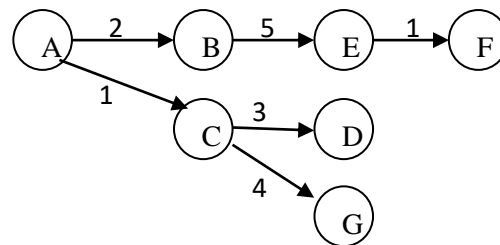
- Exemple 1 : déterminer la chaîne de distance minimale allant du sommet A au sommet F du graphe ci-dessous en utilisant l'algorithme de Dijkstra:



# Algorithme de Moore-Dijkstra (4)

- Solution de l'exemple 1

	A	B	C	D	E	F	G
A 0	0	2,A	1,A	$\infty$	$\infty$	$\infty$	$\infty$
C 1		2,A	1,A	4,C	$\infty$	$\infty$	5,C
B 2		2,A		4,C	7,B	$\infty$	5,C
D 4				4,C	7,B	$\infty$	5,C
G 5					7,B	9,G	5,C
E 7					7,B	8,E	
F 8						8,E	



Arbre de plus courts chemin à partir du noeud A

## Le routage (1)

- **Le routage** : Le routage consiste à **déterminer** la route qu'un paquet doit prendre pour atteindre une destination. Cette tâche est réalisée au niveau de la **couche RESEAU** du modèle en couches : dans cette couche, on utilise un adressage (adresses IP) qui permet de spécifier à quel réseau appartient un équipement (hôte ou routeur).
- Chaque équipement achemine un paquet en fonction de **l'adresse IP** de la destination.
- Pour déterminer la route à prendre, le routeur utilise sa table de routage qui indique pour chaque **destination**, la **route** (interface ou passerelle) à prendre : cela est appelé routage de proche en proche.



## Le routage(2)

### Principe de fonctionnement du routage.

- Lorsqu'un ordinateur **émit** un message vers un autre, **hors** de son réseau, ce message est transmis au routeur. Ce dernier effectue les actions suivantes :
  - Lire l'adresse du destinataire.
  - Consulte sa table de routage pour déterminer la route à suivre pour atteindre cette destination.
  - Transmet le message au routeur suivant (ou au destinataire s'il est à côté).
- Un routeur **construit** sa table de routage Manuellement ou automatiquement par échange de routes avec ses voisins.

## Le routage (3)

- Un routeur utilise une table de routage pour déterminer le lieu d'expédition des paquets. La table de routage contient un ensemble de routes. Chaque route décrit la passerelle utilisée par le routeur pour atteindre le réseau demandé. Une **route** possède quatre composantes principales :
  - - le réseau de destination ;
  - - le masque de sous-réseau ;
  - - l'adresse de passerelle;
  - - le coût de la route ou la mesure.

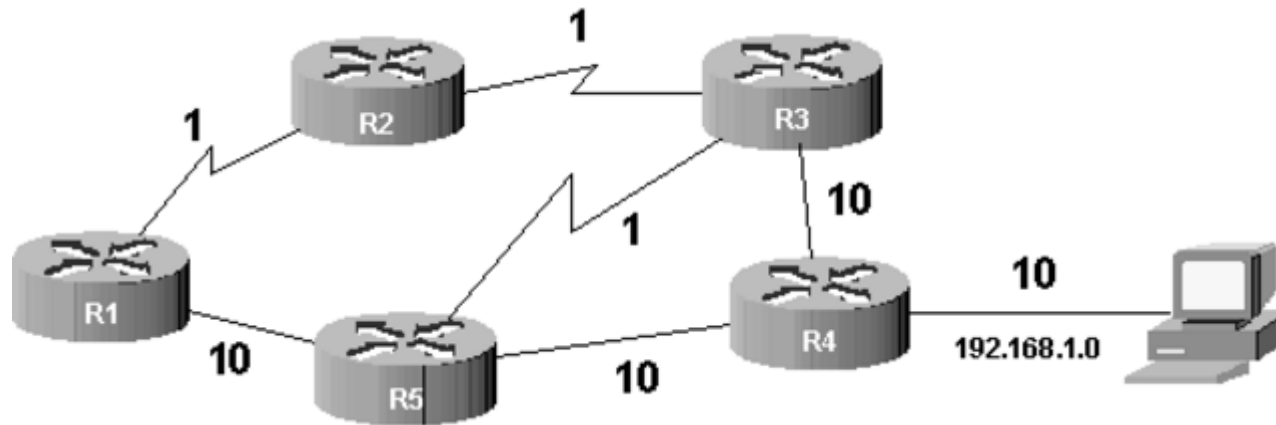
## Le routage (4)

- **Détection automatique de réseaux :**  
Concrètement, les routeurs **s'échangent** leurs tables et établissent un « **meilleur chemin** » s'il en existe plusieurs. Ce meilleur chemin dépend du **protocole** utilisé. Après la découverte initiale des réseaux, les protocoles de routage dynamique les **mettent à jour** et les gèrent dans leurs tables de routage. Les protocoles de routage dynamique déterminent également un **nouveau meilleur chemin** si le chemin initial devient **inutilisable** (ou si la topologie change).

## Application de l'algorithme de Dijkstra dans le protocole de routage: Open Shortest Path First (OSPF) (1)

- **OSPF** est un protocole de routage dynamique, Il utilise l'algorithme SPF (Shortest Path First) ou (Dijkstra), afin d'élire la meilleure route vers une destination donnée.

## Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (2)



**Notion du cout :** Supposons qu'à partir du routeur **R1**, on cherche à atteindre le réseau 192.168.1.0. Dans une telle situation, d'autres protocoles auraient élu la route passant par R5 puisque c'est la plus courte en termes de saut. Cependant, si les liens représentés sous forme d'éclairs soient «rapides» (de type FastEthernet à 100 Mbps par exemple) et que les liens représentés sous formes de segments droits soient «lents» (de type Ethernet à 10 Mbps par exemple).

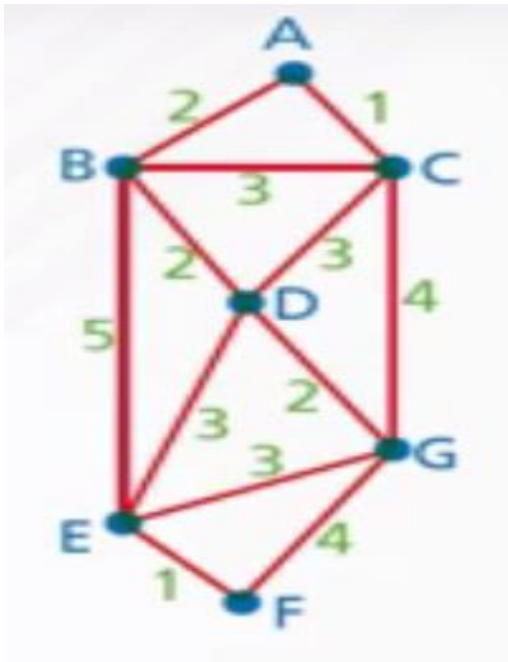
## Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (3)

- Le protocole OSPF fonctionne différemment. Il attribue un coût à chaque liaison (appelée lien) afin de privilégier l'élection de certaines routes. Plus le coût est faible, plus le lien est intéressant. Par défaut, les coûts sont déterminés en fonction de la bande passante du lien :  $\text{coût} = 10^8 / \text{bande passante de la liaison en bps}$ .
- La référence  $10^8$  correspond à un débit maximum de 100 Mbp/s.

## Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (4)

- Chaque routeur possède une **table de voisinage**, appelé Neighbor table, une **base de données de la topologie du réseau**, appelé Topology database Et une **table de routage**.
- Chaque routeur établit des relations d'adjacence avec ses voisins immédiats en envoyant des messages **hello** à intervalle régulier. Chaque routeur communique ensuite la liste des réseaux auxquels il est connecté par des messages **Link-state Advertisements (LSA)** propagés de proche en proche à tous les routeurs du réseau.
- L'ensemble des LSA forme une **base de données de l'état des liens (LSDB)** identique pour tous les routeurs du réseau. Chaque routeur utilise ensuite l'algorithme de Dijkstra, pour déterminer la route la plus courte vers chacun des réseaux connus dans la LSDB.
- En cas de changement de topologie, de nouveaux LSA sont propagés de proche en proche, et l'algorithme Dijkstra est exécuté à nouveau sur chaque routeur.

## Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (5)



Destination	Routeur suivant	Coût
B	B	2
C	C	1
D	C	4
E	B	7
F	B	8
G	C	5

Un exemple illustratif du Contenu de la table de routage du routeur A pour la topologie présentée par le graph à gauche



## Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (6)

### Le concept de zone (*area*)

- OSPF a été conçu pour supporter de nombreuses routes. Donc, afin d'éviter que la bande passante ne soit engloutie dans la diffusion des routes, OSPF introduit le concept de **zone** (*area*). Le réseau est divisé en plusieurs zones de routage qui contiennent des routeurs et des hôtes.
- L'intérêt de définir des zones est de limiter le trafic de routage, de réduire la fréquence des calculs du plus court chemin par l'algorithme SPF ainsi que d'avoir une table de routage plus petite.

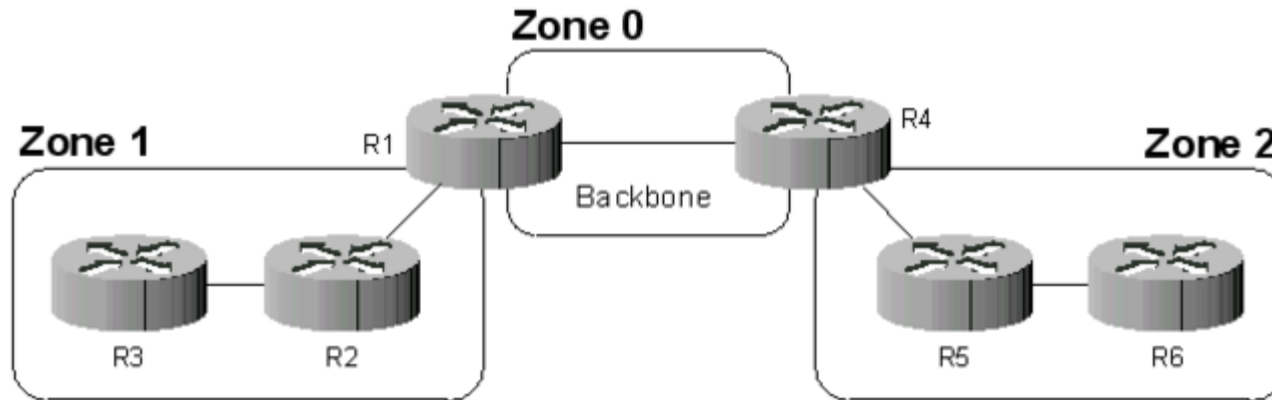
## Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (7)

### Le concept de zone (*area*)

- Chaque zone, identifiée par un numéro, possède sa propre topologie et ne connaît pas la topologie des autres zones. Chaque routeur d'une zone donnée ne connaît que les routeurs de sa propre zone ainsi que la façon d'atteindre une zone particulière, la zone numéro 0.
- Toutes les zones doivent être connectées physiquement à la zone 0 (appelée *backbone* ou réseau fédérateur). Elle est constituée de plusieurs routeurs interconnectés.
- Le *backbone* est chargé de diffuser les informations de routage qu'il reçoit d'une zone aux autres zones. Tout routage basé sur OSPF doit posséder une zone 0.

# Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (8)

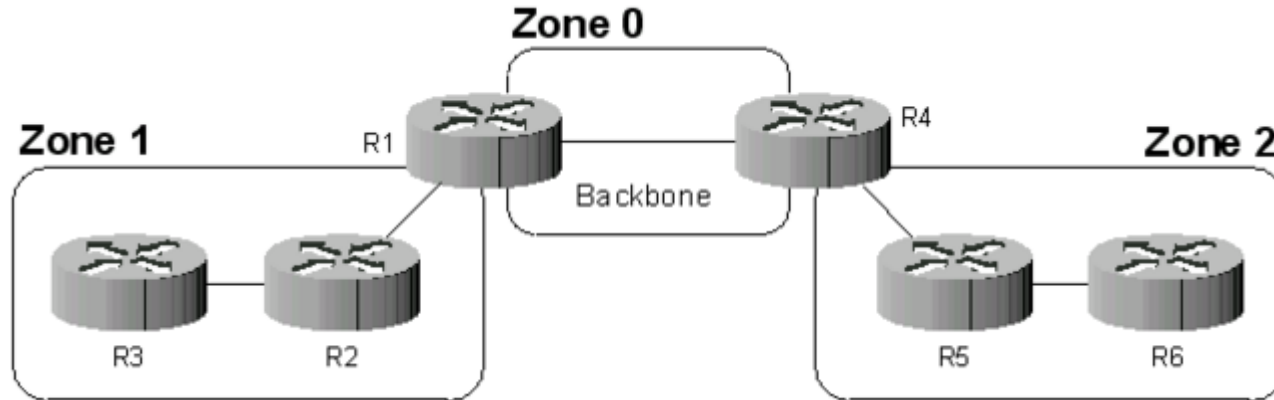
## Le concept de zone (*area*)



- le réseau de la figure est découpé en trois zones dont le *backbone*. Les routeurs de la zone 1, par exemple, ne connaissent pas les routeurs de la zone 2 et encore moins la topologie de la zone 2.

# Application de l'algorithme de Dijkstra dans le protocole de routage Open Shortest Path First (OSPF) (9)

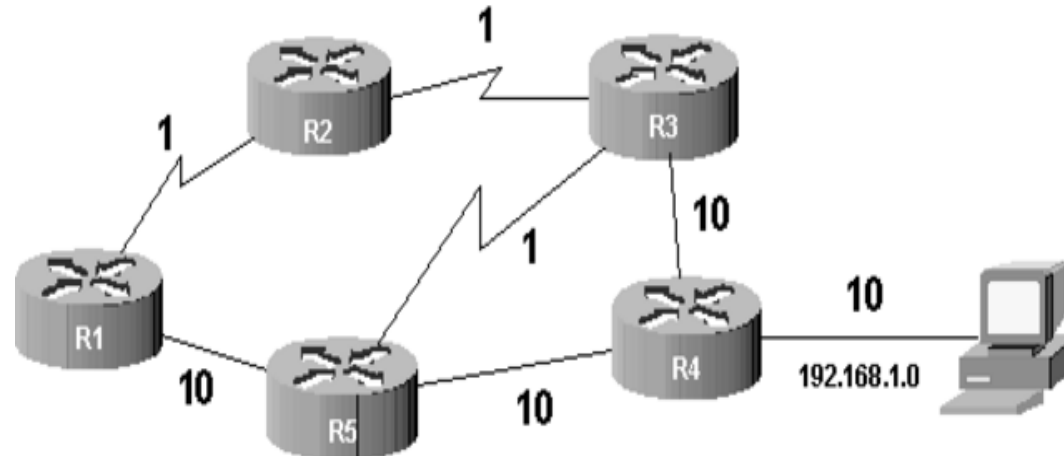
## Le concept de zone (*area*)



- Les routeurs R1 et R4 sont particuliers puisqu'ils sont entre plusieurs zones (on les appelle ABR pour *Area Border Router* ou routeur de bord). Ces routeurs maintiennent une base de données topologique pour chaque zone à laquelle ils sont connectés.
- Les ABR sont des points de sortie pour les zones ce qui signifie que les informations de routage destinées aux autres zones doivent passer par l'ABR local à la zone. L'ABR se charge alors de retransmettre les informations de routage au *backbone*.

# TP1 : OSPF

- visualiser de la table de routage (protocole OSPF) et la table de voisinage du routeur R1 de la topologie de la figure qui suit:



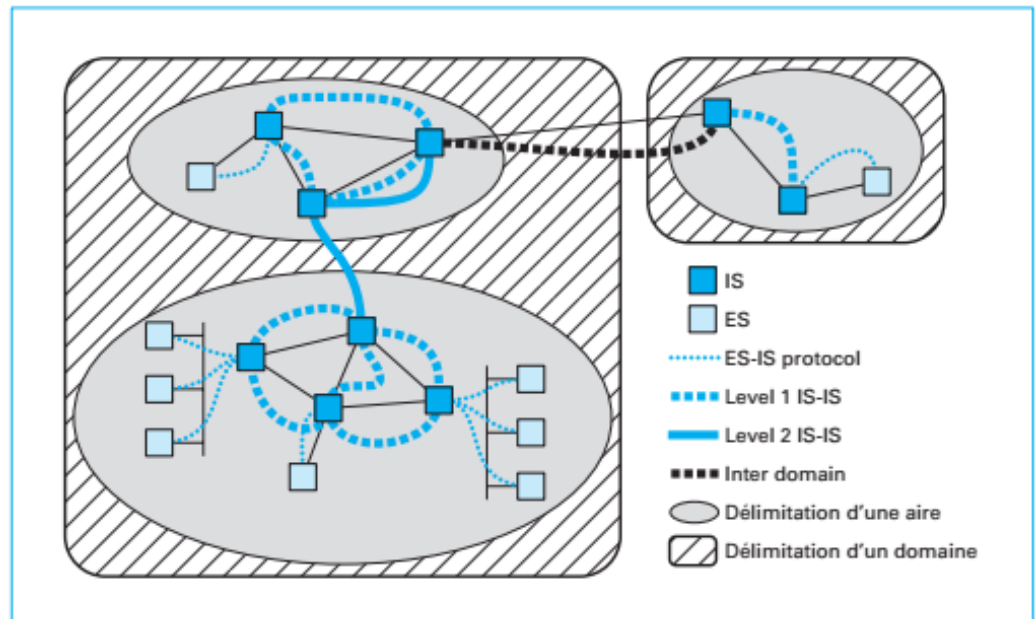
# Application de l'algorithme de Dijkstra dans le protocole de routage Intermediate System to Intermediate System(IS-IS) (1)

- IS-IS (*Intermediate System to Intermediate System*) est un protocole de routage interne à état de lien. Il a été standardisé par l'ISO. C'est un protocole de niveau 3 (contrairement à OSPF qui est de niveau 4).
- ce protocole est indépendant d'IP, que ce soit IPv4 ou IPv6.

# Application de l'algorithme de Dijkstra dans le protocole de routage Intermediate System to Intermediate System (IS-IS) (2)

- Un routeur qui implémente IS-IS peut être soit :
- level-1 (routage intra aire),
- level-2 (routage inter aire),
- level-1-2 (routage intra et inter aire).

Un routeur de niveau 1 n'a de voisins que dans son aire, alors qu'un routeur de niveau 2 peut avoir des voisins dans une autre aire. Il n'y a pas d'aire de backbone (contrairement à OSPF). Le backbone est constitué de la réunion de tous les routeurs de level-2.



# Application de l'algorithme de Dijkstra dans le protocole de routage Intermediate System to Intermediate System(IS-IS) (3)

- **Comparaison**
- Les deux protocoles sont des (IGP:Interior gateway protocol)
- Ils distribuent les informations de routage entre les routeurs appartenant à un seul (AS : autonomous system)
- Avec support de:
  - Variable Subnet Length Masking (VLSM)
  - Authentication
  - Multi-path