

### Quelques dates clés de l'histoire d'internet

- Debut des années 1960 : idée de la création d'un réseau informatique global permettant d'interconnecter de multiples sous-réseaux.

### Quelques dates clés de l'histoire d'internet

- Début des années 1960 : idée de la création d'un réseau informatique global permettant d'interconnecter de multiples sous-réseaux.
- Début des années 1970 : naissance d'ARPANET, ancêtre d'internet.

### Quelques dates clés de l'histoire d'internet

- Début des années 1960 : idée de la création d'un réseau informatique global permettant d'interconnecter de multiples sous-réseaux.
- Début des années 1970 : naissance d'ARPANET, ancêtre d'internet.
- 1973 : définition des protocoles TCP (Transmission Control Protocol) et IP (Internet Protocol)

### Quelques dates clés de l'histoire d'internet

- Début des années 1960 : idée de la création d'un réseau informatique global permettant d'interconnecter de multiples sous-réseaux.
- Début des années 1970 : naissance d'ARPANET, ancêtre d'internet.
- 1973 : définition des protocoles TCP (Transmission Control Protocol) et IP (Internet Protocol)
- 1983 : premier serveur de noms de domaines (DNS pour domain name server))

### Quelques dates clés de l'histoire d'internet

- Début des années 1960 : idée de la création d'un réseau informatique global permettant d'interconnecter de multiples sous-réseaux.
- Début des années 1970 : naissance d'ARPANET, ancêtre d'internet.
- 1973 : définition des protocoles TCP (Transmission Control Protocol) et IP (Internet Protocol)
- 1983 : premier serveur de noms de domaines (DNS pour domain name server))
- 1989 : Naissance du *web*

### Quelques dates clés de l'histoire d'internet

- Début des années 1960 : idée de la création d'un réseau informatique global permettant d'interconnecter de multiples sous-réseaux.
- Début des années 1970 : naissance d'ARPANET, ancêtre d'internet.
- 1973 : définition des protocoles TCP (Transmission Control Protocol) et IP (Internet Protocol)
- 1983 : premier serveur de noms de domaines (DNS pour domain name server))
- 1989 : Naissance du *web*
- 2010 : Plus de 5 milliards de machines connectés.

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

Transport

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

Transport

Réseau

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

Transport

Réseau

Liaison

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

HTTP, POP, FTP .....

Transport

Réseau

Liaison

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

HTTP, POP, FTP .....

Transport

TCP, UDP

Réseau

Liaison

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

HTTP, POP, FTP .....

Transport

TCP, UDP

Réseau

IPv4, IPv6

Liaison

### Protocoles TCP/IP

Internet fonctionne suivant une architecture réseau en 4 couches :

Applications

HTTP, POP, FTP .....

Transport

TCP, UDP

Réseau

IPv4, IPv6

Liaison

Ethernet, WIFI

## C12 Protocoles de routage

### Remarques

- Chaque couche ne communique qu'avec les couches voisines.

## C12 Protocoles de routage

### Remarques

- Chaque couche ne communique qu'avec les couches voisines.
- Chaque couche ajoute les données dont il a besoin pour fonctionner à l'information transmise. C'est ce qu'on appelle l'**encapsulation** des données.

### Remarques

- Chaque couche ne communique qu'avec les couches voisines.
- Chaque couche ajoute les données dont il a besoin pour fonctionner à l'information transmise. C'est ce qu'on appelle l'**encapsulation** des données.
- Un autre modèle en 7 couches existe : le modèle OSI.

### Adresse mac

- On considère un réseau local, de deux ordinateurs (reliés par un simple cable) ou de plusieurs ordinateurs (reliés par un *switch*)

### Adresse mac

- On considère un réseau local, de deux ordinateurs (reliés par un simple câble) ou de plusieurs ordinateurs (reliés par un *switch*)
- Les adresses MAC (pour *Media Access Control*) permettent d'identifier de façon unique un élément du réseau. Elle sont composées de six nombres en hexadécimal séparé par le caractère deux points ( : ), par exemple 1A:B2:EC:AE:B0:DE.

### Adresse mac

- On considère un réseau local, de deux ordinateurs (reliés par un simple câble) ou de plusieurs ordinateurs (reliés par un *switch*)
- Les adresses *MAC* (pour *Media Access Control*) permettent d'identifier de façon unique un élément du réseau. Elle sont composées de six nombres en hexadécimal séparé par le caractère deux points ( : ), par exemple 1A:B2:EC:AE:B0:DE.
- Le protocole *arp* permet d'associer l'adresse *MAC* à l'adresse *IP*.

### Adresse mac

- On considère un réseau local, de deux ordinateurs (reliés par un simple câble) ou de plusieurs ordinateurs (reliés par un *switch*)
- Les adresses *MAC* (pour *Media Access Control*) permettent d'identifier de façon unique un élément du réseau. Elle sont composées de six nombres en hexadécimal séparé par le caractère deux points ( : ), par exemple 1A:B2:EC:AE:B0:DE.
- Le protocole *arp* permet d'associer l'adresse *MAC* à l'adresse *IP*.
- Une commande *ping* entre deux de ces machines, commence donc par un appel au protocole *arp* afin d'obtenir l'adresse *MAC* de la machine cible.

### Adresse ip

- Deux machines ne peuvent communiquer que si **elles sont sur le même réseau**, c'est à dire que leur adresse IP débute par une partie commune. La longueur de cette partie commune dépend du masque de sous réseau.

### Adresse ip

- Deux machines ne peuvent communiquer que si **elles sont sur le même réseau**, c'est à dire que leur adresse IP débute par une partie commune. La longueur de cette partie commune dépend du masque de sous réseau.
- Par exemple si le masque de sous réseau est 255.255.255.0, cela signifie que les trois premiers octets de l'adresse IP doivent être communs.

### Adresse ip

- Deux machines ne peuvent communiquer que si **elles sont sur le même réseau**, c'est à dire que leur adresse IP débute par une partie commune. La longueur de cette partie commune dépend du masque de sous réseau.
- Par exemple si le masque de sous réseau est 255.255.255.0, cela signifie que les trois premiers octets de l'adresse IP doivent être communs.
- Un masque de sous réseau se note aussi en notation CIDR (pour Classless Inter-Domain Routing) en indiquant simplement le nombre de bits communs. Comme 3 octets = 24 bits, Le masque précédent se note donc plus simplement /24.

### Adresse ip

- Deux machines ne peuvent communiquer que si **elles sont sur le même réseau**, c'est à dire que leur adresse IP débute par une partie commune. La longueur de cette partie commune dépend du masque de sous réseau.
- Par exemple si le masque de sous réseau est 255.255.255.0, cela signifie que les trois premiers octets de l'adresse IP doivent être communs.
- Un masque de sous réseau se note aussi en notation CIDR (pour Classless Inter-Domain Routing) en indiquant simplement le nombre de bits communs. Comme 3 octets = 24 bits, Le masque précédent se note donc plus simplement /24.
- Si le masque de sous réseau est 255.255.255.0, le nombre maximal de machines sur le réseau est de 254 (parmi les 256 possibilités, deux sont réservées. L'une pour le *broadcast* (envoi à tout le réseau) et l'autre pour le sous-réseau lui-même.

## C12 Protocoles de routage

### Exemple

On considère trois machines : A (192.168.130.10), B (192.168.155.100) et C (192.168.144.203) et on suppose qu'on a défini pour ces machines le masque de sous réseau 255.255.240.0

- Donner ce masque de sous réseau en notation CIDR.

### Exemple

On considère trois machines : A (192.168.130.10), B (192.168.155.100) et C (192.168.144.203) et on suppose qu'on a défini pour ces machines le masque de sous réseau 255.255.240.0

- Donner ce masque de sous réseau en notation CIDR.

### Exemple

On considère trois machines : A (192.168.130.10), B (192.168.155.100) et C (192.168.144.203) et on suppose qu'on a défini pour ces machines le masque de sous réseau 255.255.240.0

- Donner ce masque de sous réseau en notation CIDR.

### Exemple

On considère trois machines : A (192.168.130.10), B (192.168.155.100) et C (192.168.144.203) et on suppose qu'on a défini pour ces machines le masque de sous réseau 255.255.240.0

- Donner ce masque de sous réseau en notation CIDR.  
On convertit 240 en binaire :  $(240)_{10} = (11110000)_2$ , il y a donc 20 bits (8+8+4) en commun le masque est donc /20.
- Indiquer les machines qui peuvent communiquer entre elles.

### Exemple

On considère trois machines : A (192.168.130.10), B (192.168.155.100) et C (192.168.144.203) et on suppose qu'on a défini pour ces machines le masque de sous réseau 255.255.240.0

- Donner ce masque de sous réseau en notation CIDR.  
On convertit 240 en binaire :  $(240)_{10} = (11110000)_2$ , il y a donc 20 bits (8+8+4) en commun le masque est donc /20.
- Indiquer les machines qui peuvent communiquer entre elles.  
Le problème porte sur le troisième octet, les adresses IP doivent avoir leur 4 premier octet communs pour pouvoir communiquer :

### Exemple

On considère trois machines : A (192.168.130.10), B (192.168.155.100) et C (192.168.144.203) et on suppose qu'on a défini pour ces machines le masque de sous réseau 255.255.240.0

- Donner ce masque de sous réseau en notation CIDR.

On convertit 240 en binaire :  $(240)_{10} = (11110000)_2$ , il y a donc 20 bits (8+8+4) en commun le masque est donc /20.

- Indiquer les machines qui peuvent communiquer entre elles.

Le problème porte sur le troisième octet, les adresses IP doivent avoir leur 4 premier octet communs pour pouvoir communiquer :

- A : 192.168.130.10, or  $(130)_{10} = (100000010)_2$
- B : 192.168.155.100, or  $(155)_{10} = (10011011)_2$
- C : 192.168.144.203, or  $(144)_{10} = (10010000)_2$

B et C peuvent communiquer entre elles mais pas avec A.

- Confirmer éventuellement par un test dans Filius

### Routeurs

- Les routeurs permettent de faire communiquer des ordinateurs appartenant à des sous réseaux différents.

### Routeurs

- Les routeurs permettent de faire communiquer des ordinateurs appartenant à des sous réseaux différents.
- A titre d'exemple une *box internet* dans une maison joue le rôle de switch (elle permet aux différents ordinateurs du foyer de communiquer entre eux) mais aussi de routeur (elle permet de communiquer avec des ordinateurs hors de la maison).

## Routeurs

- Les routeurs permettent de faire communiquer des ordinateurs appartenant à des sous réseaux différents.
- A titre d'exemple une *box internet* dans une maison joue le rôle de switch (elle permet aux différents ordinateurs du foyer de communiquer entre eux) mais aussi de routeur (elle permet de communiquer avec des ordinateurs hors de la maison).
- Les *tables de routage* sont des informations stockés localement dans chaque routeur et lui permettant d'orienter les paquets qu'ils reçoit vers un autre routeur ou un sous réseau avec lequel il communique.

### Protocoles de routage

- Pour des réseaux de petites tailles, les tables de routage peuvent être écrites « à la main ».

### Protocoles de routage

- Pour des réseaux de petites tailles, les tables de routage peuvent être écrites « à la main ». Les inconvénients sont nombreux car les tables sont alors statiques et ne s'adaptent pas (par exemple lors d'une panne ou de l'ajout de nouveaux éléments au réseau).

### Protocoles de routage

- Pour des réseaux de petites tailles, les tables de routage peuvent être écrites « à la main ». Les inconvénients sont nombreux car les tables sont alors statiques et ne s'adaptent pas (par exemple lors d'une panne ou de l'ajout de nouveaux éléments au réseau).
- Des protocoles de routage permettant de générer de façon automatique les tables de routage ont donc été mis au point, nous allons en présenter deux :

### Protocoles de routage

- Pour des réseaux de petites tailles, les tables de routage peuvent être écrites « à la main ». Les inconvénients sont nombreux car les tables sont alors statiques et ne s'adaptent pas (par exemple lors d'une panne ou de l'ajout de nouveaux éléments au réseau).
- Des protocoles de routage permettant de générer de façon automatique les tables de routage ont donc été mis au point, nous allons en présenter deux :
  - Le protocole de routage RIP (Routing Information Protocol) transmet les paquets de proche en proche en essayant de minimiser le nombre de routeurs traversés.

### Protocoles de routage

- Pour des réseaux de petites tailles, les tables de routage peuvent être écrites « à la main ». Les inconvénients sont nombreux car les tables sont alors statiques et ne s'adaptent pas (par exemple lors d'une panne ou de l'ajout de nouveaux éléments au réseau).
- Des protocoles de routage permettant de générer de façon automatique les tables de routage ont donc été mis au point, nous allons en présenter deux :
  - Le protocole de routage RIP (Routing Information Protocol) transmet les paquets de proche en proche en essayant de minimiser le nombre de routeurs traversés.
  - Le protocole de routage OSPF (Open Shortest Path First) transmet les paquets en tenant compte de la vitesse de connection entre les routeurs

### Le protocole RIP

- A l'origine les routeurs ne connaissent que les réseaux auxquels ils sont directement connectés

### Le protocole RIP

- A l'origine les routeurs ne connaissent que les réseaux auxquels ils sont directement connectés
- Puis régulièrement (toutes les 30 secondes), chaque routeur reçoit la table de routage de ses voisins et il met alors à jour sa propre table :

### Le protocole RIP

- A l'origine les routeurs ne connaissent que les réseaux auxquels ils sont directement connectés
- Puis régulièrement (toutes les 30 secondes), chaque routeur reçoit la table de routage de ses voisins et il met alors à jour sa propre table :
  - S'il y découvre une route vers un réseau inconnu, il l'ajoute à sa propre table en augmentant de 1 la distance.

### Le protocole RIP

- A l'origine les routeurs ne connaissent que les réseaux auxquels ils sont directement connectés
- Puis régulièrement (toutes les 30 secondes), chaque routeur reçoit la table de routage de ses voisins et il met alors à jour sa propre table :
  - S'il y découvre une route vers un réseau inconnu, il l'ajoute à sa propre table en augmentant de 1 la distance.
  - S'il y découvre une route vers un réseau connu mais plus courte, il met à jour sa table.

### Le protocole RIP

- A l'origine les routeurs ne connaissent que les réseaux auxquels ils sont directement connectés
- Puis régulièrement (toutes les 30 secondes), chaque routeur reçoit la table de routage de ses voisins et il met alors à jour sa propre table :
  - S'il y découvre une route vers un réseau inconnu, il l'ajoute à sa propre table en augmentant de 1 la distance.
  - S'il y découvre une route vers un réseau connu mais plus courte, il met à jour sa table.
- Si un routeur ne reçoit pas d'information d'un voisin figurant dans sa table (pendant 3 min), il considère celui ci comme en panne (et lui affecte la distance maximale de 16).

### Inconvénients du protocole RIP

- L'échange régulier des tables de routage augmente la quantité d'information qui circule sur le réseau

### Inconvénients du protocole RIP

- L'échange régulier des tables de routage augmente la quantité d'information qui circule sur le réseau
- Le protocole se limite au réseau de petites tailles car une distance maximale de 15 sauts est imposée

## C12 Protocoles de routage

### Inconvénients du protocole RIP

- L'échange régulier des tables de routage augmente la quantité d'information qui circule sur le réseau
- Le protocole se limite au réseau de petites tailles car une distance maximale de 15 sauts est imposée
- Le protocole se base uniquement sur le nombre de sauts et pas sur la qualité de la liaison entre deux routeurs qui peut être très différentes :

Technologie	BP descendante	BP montante
Bluetooth	3 Mbit/s	3 Mbit/s
Ethernet	10 Mbit/s	10 Mbit/s
Wi-Fi	10 Mbit/s 10 Gbits/s	10 Mbit/s 10 Gbits/s
ADSL	13 Mbit/s	1 Mbit/s
4G	100 Mbit/s	50 Mbit/s
Satellite	50 Mbit/s	1 Mbit/s
Fast Ethernet	100 Mbit/s	100 Mbit/s
FFTH (fibre)	10 Gbit/s	10 Gbit/s
5G	20 Gbit/s	10 Gbit/s

### Protocoles OSPF

- On commence par attribuer un coût *inversement proportionnel à la qualité de la connection* à chaque liaison.

### Protocoles OSPF

- On commence par attribuer un coût *inversement proportionnel à la qualité de la connection* à chaque liaison.

Cette formule qui calcule le coût  $c$  d'une liaison ayant un débit  $d$  (en bit/s)

peut être par exemple :

### Protocoles OSPF

- On commence par attribuer un coût *inversement proportionnel* à la *qualité de la connection* à chaque liaison.

Cette formule qui calcule le coût  $c$  d'une liaison ayant un débit  $d$  (en bit/s) peut être par exemple :  $c = \frac{10^8}{d}$ . Mais elle peut-être différente et sera donné à chaque exercice.

### Protocoles OSPF

- On commence par attribuer un coût *inversement proportionnel* à la *qualité de la connection* à chaque liaison.  
Cette formule qui calcule le coût  $c$  d'une liaison ayant un débit  $d$  (en bit/s) peut être par exemple :  $c = \frac{10^8}{d}$ . Mais elle peut-être différente et sera donné à chaque exercice.
- On transforme donc le réseau en un **graphe pondéré** dans lequel on doit rechercher le plus court chemin entre deux sommets.

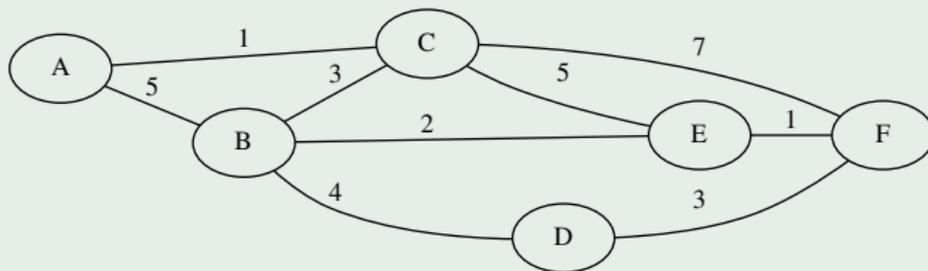
### Protocoles OSPF

- On commence par attribuer un coût *inversement proportionnel* à la *qualité de la connection* à chaque liaison.  
Cette formule qui calcule le coût  $c$  d'une liaison ayant un débit  $d$  (en bit/s) peut être par exemple :  $c = \frac{10^8}{d}$ . Mais elle peut-être différente et sera donné à chaque exercice.
- On transforme donc le réseau en un **graphe pondéré** dans lequel on doit rechercher le plus court chemin entre deux sommets.
  - Dans les cas simples de petits graphes, la réponse est intuitive.

### Protocoles OSPF

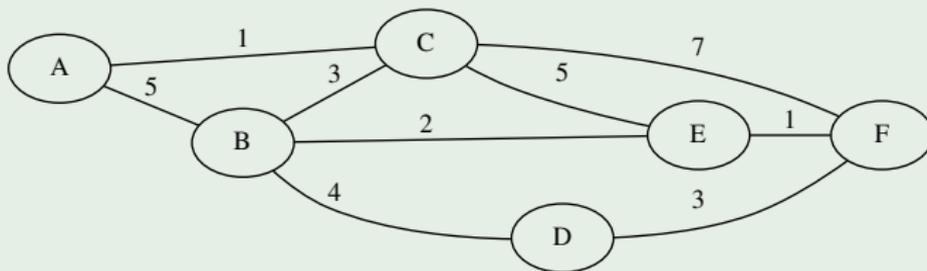
- On commence par attribuer un coût *inversement proportionnel* à la *qualité de la connection* à chaque liaison.  
Cette formule qui calcule le coût  $c$  d'une liaison ayant un débit  $d$  (en bit/s) peut être par exemple :  $c = \frac{10^8}{d}$ . Mais elle peut-être différente et sera donné à chaque exercice.
- On transforme donc le réseau en un **graphe pondéré** dans lequel on doit rechercher le plus court chemin entre deux sommets.
  - Dans les cas simples de petits graphes, la réponse est intuitive.
  - Sinon, on dispose d'un algorithme permettant de répondre à ce problème : l'algorithme de **Dijkstra** (informaticien néerlandais, 1930–2002).

## Algorithme de Dijkstra



A	B	C	D	E	F	

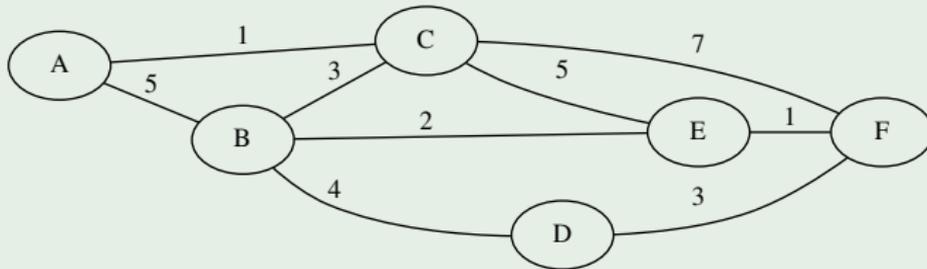
## Algorithme de Dijkstra



A	B	C	D	E	F	
0 (A)						

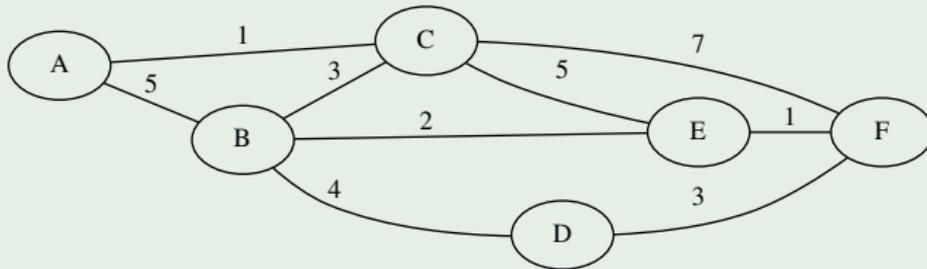
# C12 Protocoles de routage

## Algorithme de Dijkstra



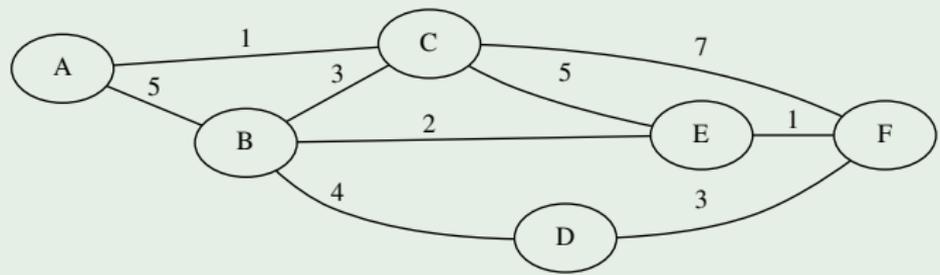
A	B	C	D	E	F	
0 (A)						A
✓						
✓						
✓						
✓						
✓						

## Algorithme de Dijkstra



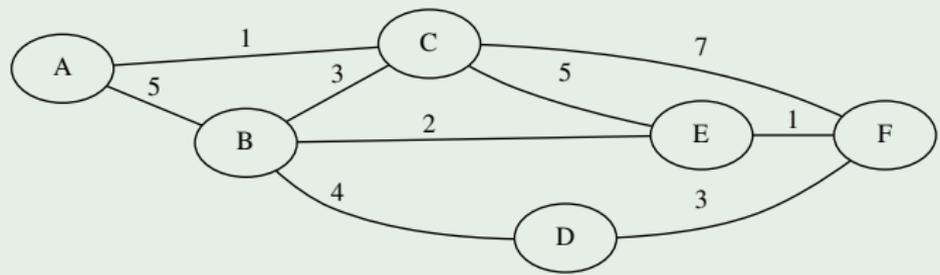
A	B	C	D	E	F	
0 (A)	5 (A)					A
✓						
✓						
✓						
✓						
✓						

## Algorithme de Dijkstra



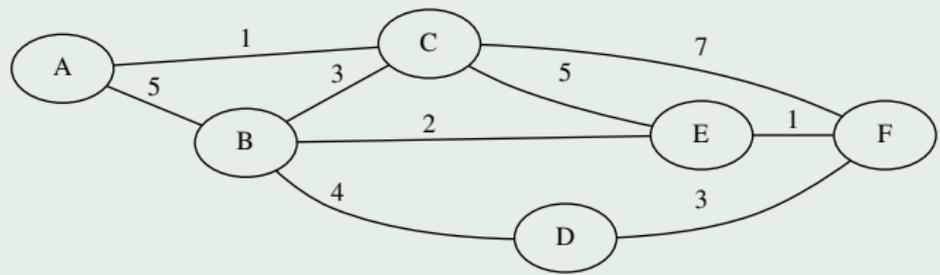
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓						
✓						
✓						
✓						
✓						

## Algorithme de Dijkstra



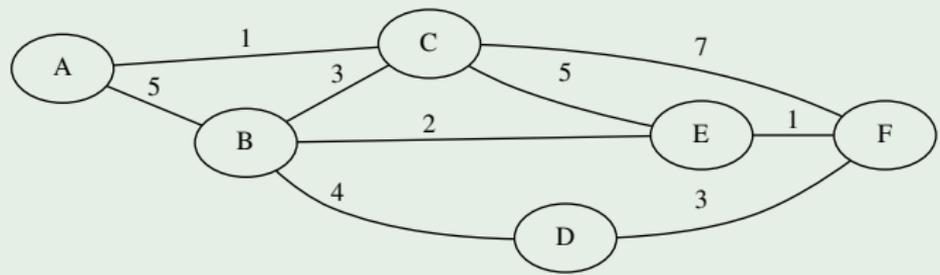
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓		1 (A)				
✓						
✓						
✓						
✓						

## Algorithme de Dijkstra



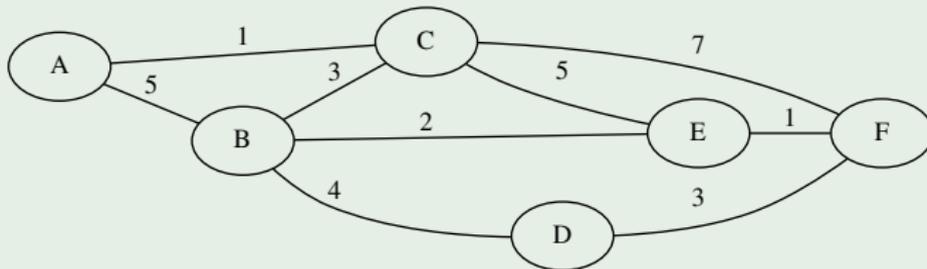
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓		1 (A)				C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

## Algorithme de Dijkstra



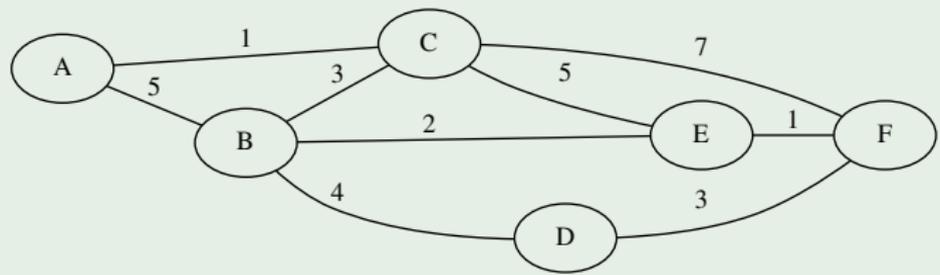
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)				C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

## Algorithme de Dijkstra



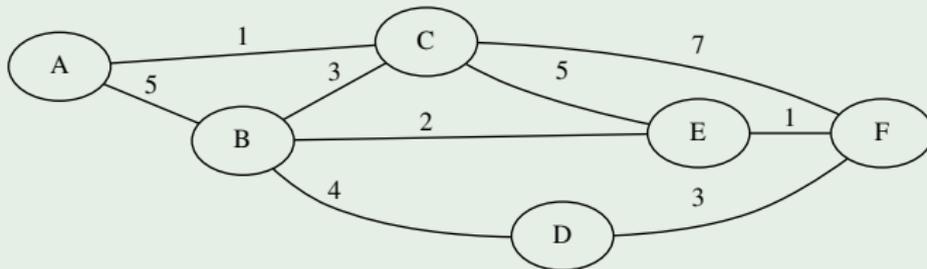
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)		C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

## Algorithme de Dijkstra



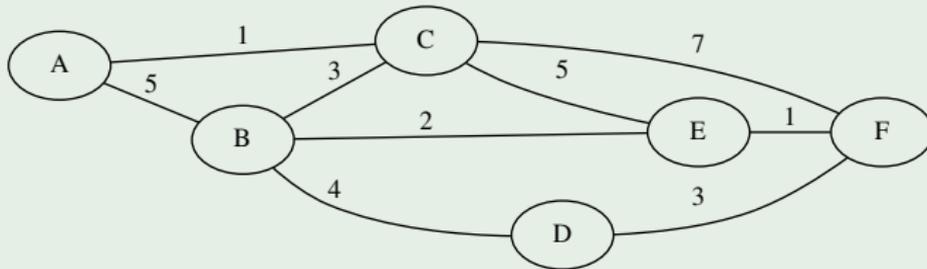
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

## Algorithme de Dijkstra



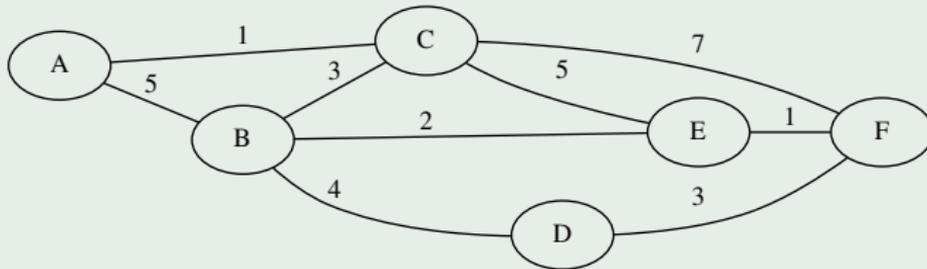
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓				
✓		✓				
✓		✓				
✓		✓				

## Algorithme de Dijkstra



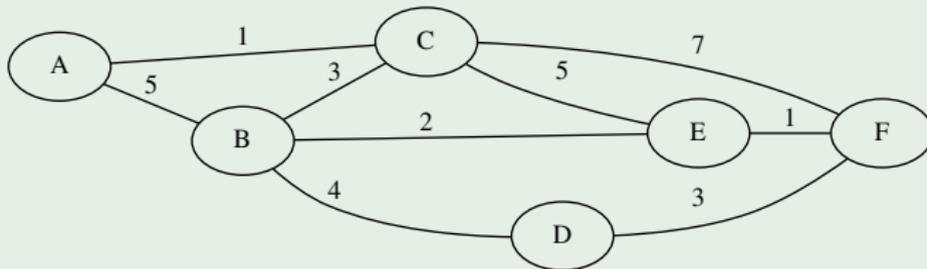
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓				B
✓	✓	✓				
✓	✓	✓				
✓	✓	✓				

## Algorithme de Dijkstra



A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)			B
✓	✓	✓				
✓	✓	✓				
✓	✓	✓				

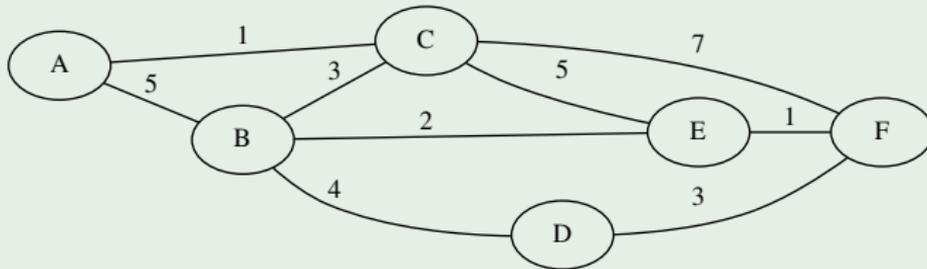
## Algorithme de Dijkstra



A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓				
✓	✓	✓				
✓	✓	✓				

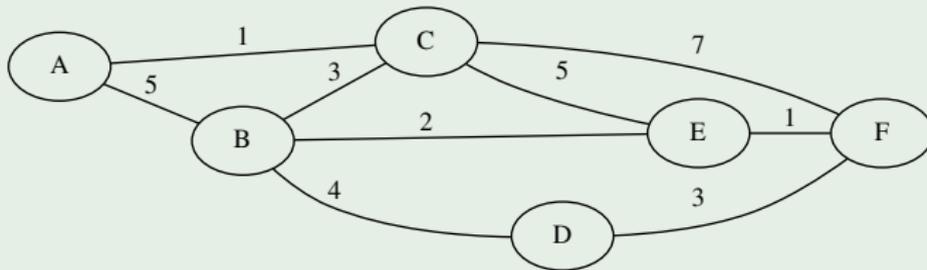
# C12 Protocoles de routage

## Algorithme de Dijkstra



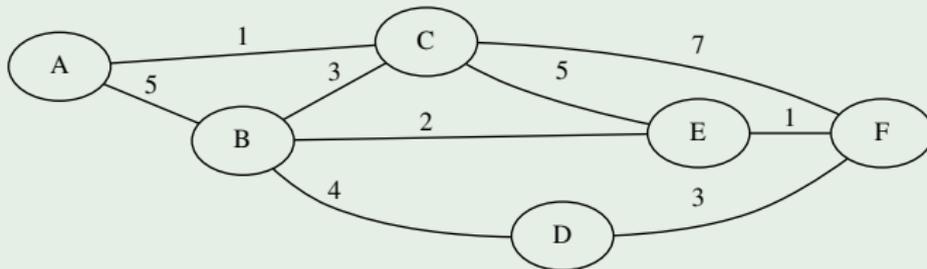
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓		6 (B)		
✓	✓	✓				
✓	✓	✓				

## Algorithme de Dijkstra



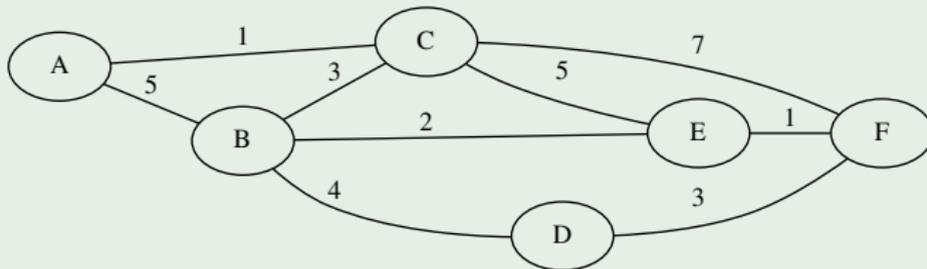
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓		6 (B)		E
✓	✓	✓		✓		
✓	✓	✓		✓		

## Algorithme de Dijkstra



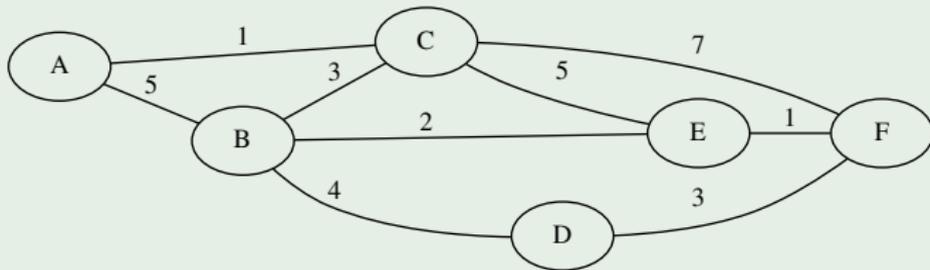
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓		6 (B)	7 (E)	E
✓	✓	✓		✓		
✓	✓	✓		✓		

## Algorithme de Dijkstra



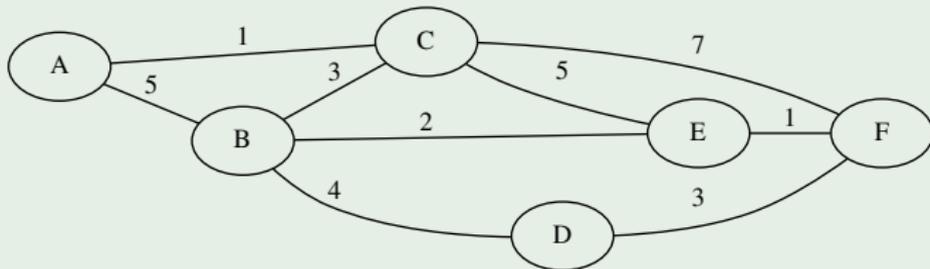
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓	8 (B)	6 (B)	7 (E)	E
✓	✓	✓		✓		
✓	✓	✓		✓		

## Algorithme de Dijkstra



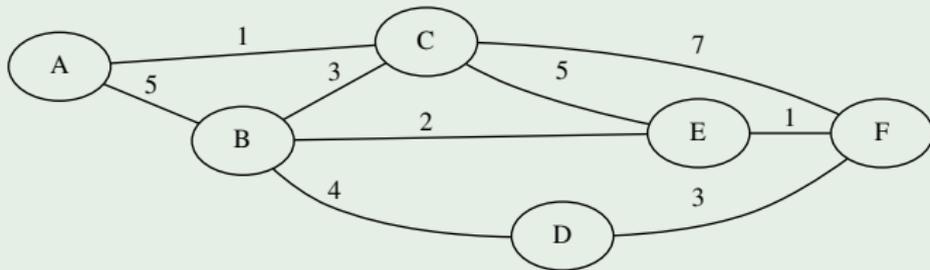
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓	8 (B)	6 (B)	7 (E)	E
✓	✓	✓		✓	7 (E)	
✓	✓	✓		✓	✓	

## Algorithme de Dijkstra



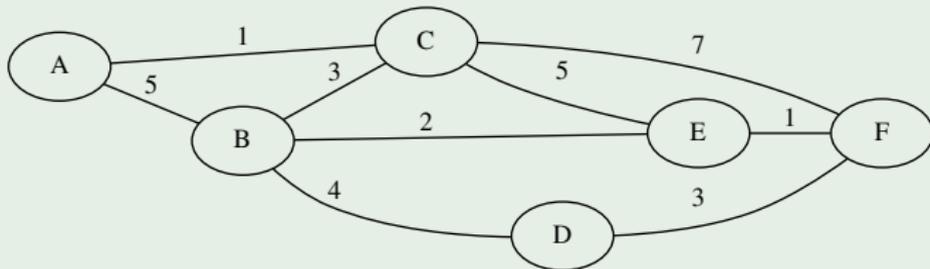
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓	8 (B)	6 (B)	7 (E)	E
✓	✓	✓		✓	7 (E)	F
✓	✓	✓		✓	✓	

## Algorithme de Dijkstra



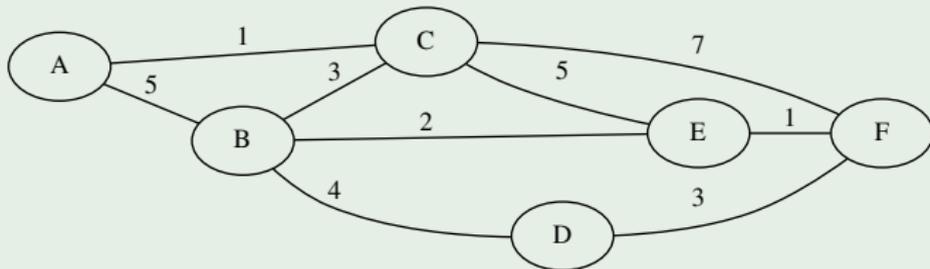
A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓	8 (B)	6 (B)	7 (E)	E
✓	✓	✓	8 (B)	✓	7 (E)	F
✓	✓	✓		✓	✓	

## Algorithme de Dijkstra



A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓	8 (B)	6 (B)	7 (E)	E
✓	✓	✓	8 (B)	✓	7 (E)	F
✓	✓	✓	8 (B)	✓	✓	

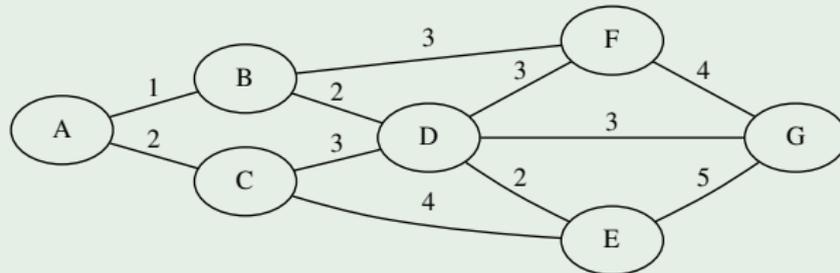
## Algorithme de Dijkstra



A	B	C	D	E	F	
0 (A)	5 (A)	1 (A)				A
✓	4 (C)	1 (A)		6 (C)	8 (C)	C
✓	4 (C)	✓	8 (B)	6 (B)	8 (C)	B
✓	✓	✓	8 (B)	6 (B)	7 (E)	E
✓	✓	✓	8 (B)	✓	7 (E)	F
✓	✓	✓	8 (B)	✓	✓	D

# C12 Protocoles de routage

## Algorithme de Dijkstra



A	B	C	D	E	F	G	