

# Couche internet (routage)

La couche internet qui soutient la couche de transport sert à faire acheminer les informations d'une machine source vers une machine destination. Cela se fait cependant sans garantie de fiabilité, c'est pour cela que le protocole TCP est nécessaire.

Certains systèmes, principalement ceux qui transitaient sur le réseau téléphonique, nécessitent une phase d'ouverture de connexion. Ce n'est cependant plus obligatoire aujourd'hui.

## Identification des machines

Les machines sont identifiées par des adresses sur 32 bits (IPv4) ou 128 bits (IPv6).

Ces adresses sont écrites généralement sous forme décimale pointée. On regroupe donc les adresses par octets (8 bits) que l'on représente sous forme décimale (pour l'IPv4).

Par exemple :

```
11000001 10111110 01000000 01111100
193      .190      .64      .124
```

En IPv6, les adresses sont codées sur 128 bits, sont représentées en hexadécimal par blocs de 16 bits séparés par `:`. On peut également abrégier les `0` consécutifs en utilisant `::`

Ainsi `2001:0bc8:38eb:fe10:0000:0000:0000:0011` devient simplement `2001:0bc8:38eb:fe10::11`

## Sous-réseaux

Pour pouvoir se connecter directement à une autre machine, il faut que cette dernière se situe dans le même **sous-réseau**. Chaque sous-réseau est lui-même identifié par une adresse IPv4 particulière.

Pour savoir si machines sont directement connectées sur un sous-réseau donné. Il faut appliquer un **masque de sous-réseau** sur l'**IP du réseau**.

```
IP RES : 11000000.10101000.00000001.00000010 - 192.168.1.2
MASQUE : 11111111.11111111.11111111.00000000 - 255.255.255.0
-----
PREFIX : 11000000.10101000.00000001.00000000 - 192.168.1.0
```

Cela signifie qu'il y a 256 adresses possibles ( $2^{(32 - \text{nombre de 1 dans le masque})}$ ), qui auront toutes un certain préfixe défini plus tôt. Par exemple, 192.168.1.1, 192.168.1.5, 192.168.1.255 sont toutes des adresses faisant partie d'un seul et même sous-réseau 192.168.1.2.

Note : le sous-réseau peut plus simplement être indiqué via la notation <IP RÉSEAU>/<NOMBRE DE 1 DU MASQUE>. Par exemple, plus tôt, on avait un sous-réseau 192.168.1.2/24. Il est aussi bon de noter qu'il y a également une adresse de **broadcast** qui permet de communiquer des paquets à tout le monde dans le réseau.

# Routage

La plupart du temps, on communique avec des machines qui sont en dehors de notre réseau direct. Il faut donc connecter les routeurs entre eux et utiliser des algorithmes pour pouvoir acheminer les informations là où il faut.

## Routeur

Pour transférer des paquets (unité d'information dans la couche internet), ces derniers transitent par des **routeurs**. Ces derniers sont des relais au niveau de la couche réseau et ont pour but de trouver le meilleur chemin pour faire transiter l'information.

Ainsi, il peut interconnecter des réseaux de natures différentes, le routeur (box) chez vous peut connecter votre réseau sur le réseau téléphonique (xDSL) ou de télédistribution (coaxial).

Le routeur a aussi plusieurs interfaces réseau avec lesquelles il communique, tel que le Wifi, une connexion au réseau de l'ISP (Internet Service Provider tel que Proximus), etc.

## Modèles de routage

Il existe deux modèles de routage de l'information :

	Datagrammes	Circuits virtuels
<b>Configuration du connexion</b>	Pas obligatoire	Obligatoire
<b>Adressage</b>	Le paquet contient les adresses source et de destination complètes	Le paquet contient l'identifiant du circuit virtuel.
<b>Informations d'état</b>	Nul autre que la table de routeur contenant le réseau de destination	Chaque numéro de circuit virtuel entré dans la table de configuration, utilisé pour le routage.
<b>Routage</b>	Paquets acheminés indépendamment	Route établie à la configuration, tous les paquets suivent la même route.
<b>Effet d'une panne de routeur</b>	Uniquement sur les paquets perdus lors d'un crash	Tous les circuits virtuels passant par un routeur défaillant sont interrompus.
<b>Contrôle de congestion</b>	Difficile car tous les paquets routés indépendamment.	Simple en pré-affectant suffisamment de buffers à chaque circuit virtuel lors de la configuration, puisque le nombre maximal de circuits est fixe.

Le mode **circuit virtuel** qui va sur base d'une information donnée par l'émetteur déterminer le chemin entre la source et la destination (donc avec phase d'ouverture), ensuite va transférer les paquets sur cette route, une fois terminé une des entités annonce une déconnexion et interrompt le circuit.

Ce mode, bien que simple à comprendre (c'est simplement un circuit), est très peu pratique pour de grands réseaux tels qu'internet, car chaque routeur devrait connaître tous les autres routeurs et si un routeur ne fonctionne plus, cela pourrait mettre en péril une grande partie du réseau.

C'est pourquoi la méthode utilisée par internet est le mode **datagramme** où chaque paquet mentionne l'adresse de destination, sur base de cette information les routeurs orientent les paquets vers la destination avec une **table de routage**. Les paquets sont donc transférés individuellement, il n'y a pas besoin de phase de connexion/configuration et le réseau peut grandir beaucoup plus facilement. Il faut toute fois noter qu'il est possible que plusieurs paquets ayant la même source et la même destination peuvent ne pas prendre le même chemin.

## Routage statique

Les routes vers les différentes destinations sont définies manuellement par l'administrateur·ice.

Cette méthode est très utilisée, car conceptuellement simple, mais ingérable pour les réseaux de taille importante et variable.

## Routage dynamique global

Le routage dynamique global consiste à faire en sorte que chaque routeur dispose d'une carte complète du réseau et déterminer manuellement les chemins les plus courts vers les destinations.

## Algorithme état liaisons

Au début, les routeurs échangent des informations entre eux pour pouvoir aider à construire une carte du réseau.

Chaque routeur connaît au départ les routes auquel il est relié et les coûts (les coûts sont calculés sur base de la vitesse de la connexion) associés.

Les routeurs publient ces informations aux autres routeurs afin de construire la carte du réseau. Chaque route est échangée dans un LSP (Link State Packet) qui comprend l'identification du routeur qui annonce ainsi que la destination et son coût associé.

Chaque LSP est à un numéro de séquence associé qui est incrémenté par la source à chaque nouvel envoi. Ainsi, les routeurs peuvent se souvenir des derniers numéros de séquence pour chaque LSP afin de ne pas renvoyer les mêmes LSP en boucle dans le réseau.

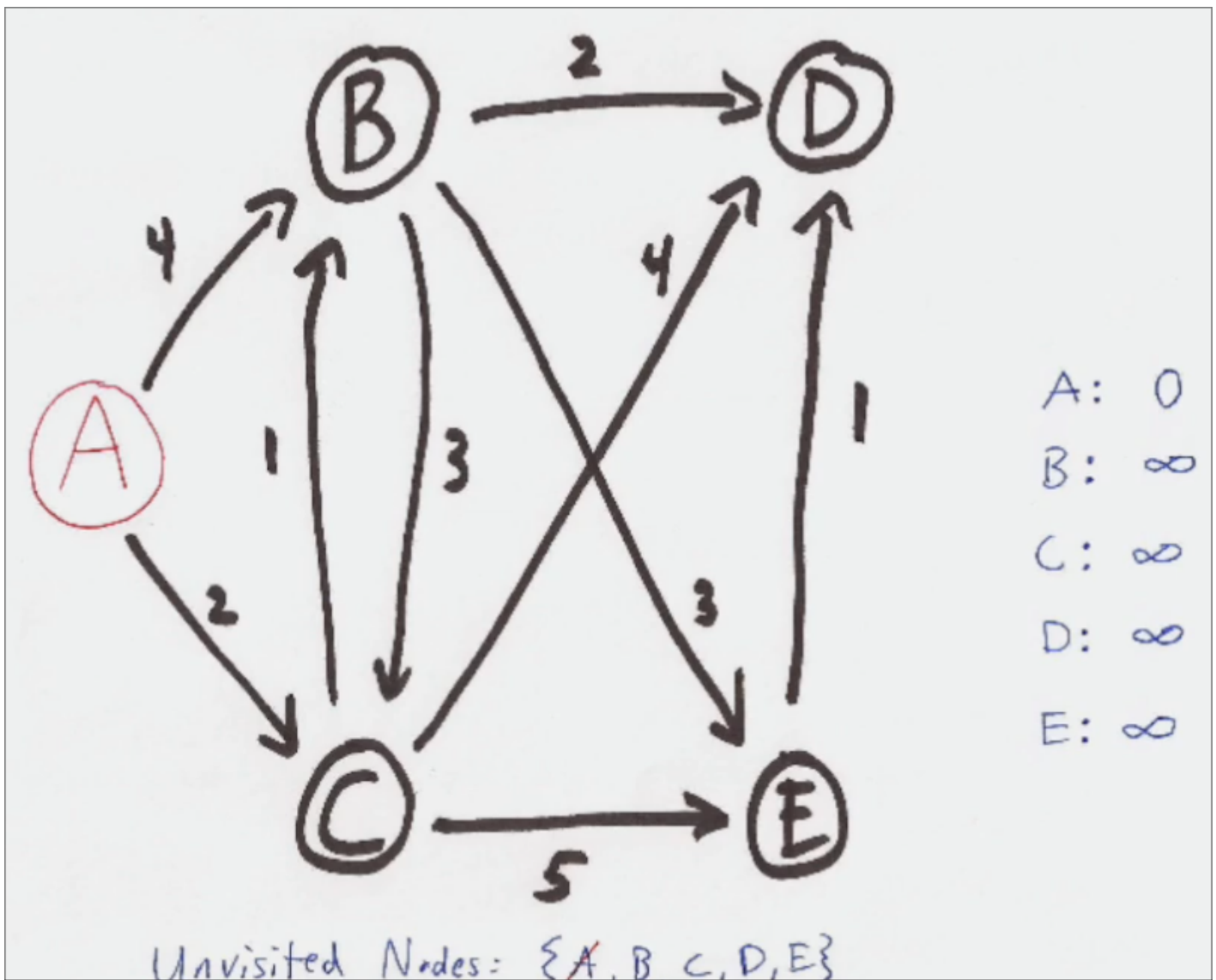
Une fois que la carte du réseau est établie sur base de ces envois, pour chaque requête, on établit le chemin le plus court entre le routeur actuel et la destination. Pour ce faire, on utilise l'**algorithme de Dijkstra**.

L'algorithme de Dijkstra

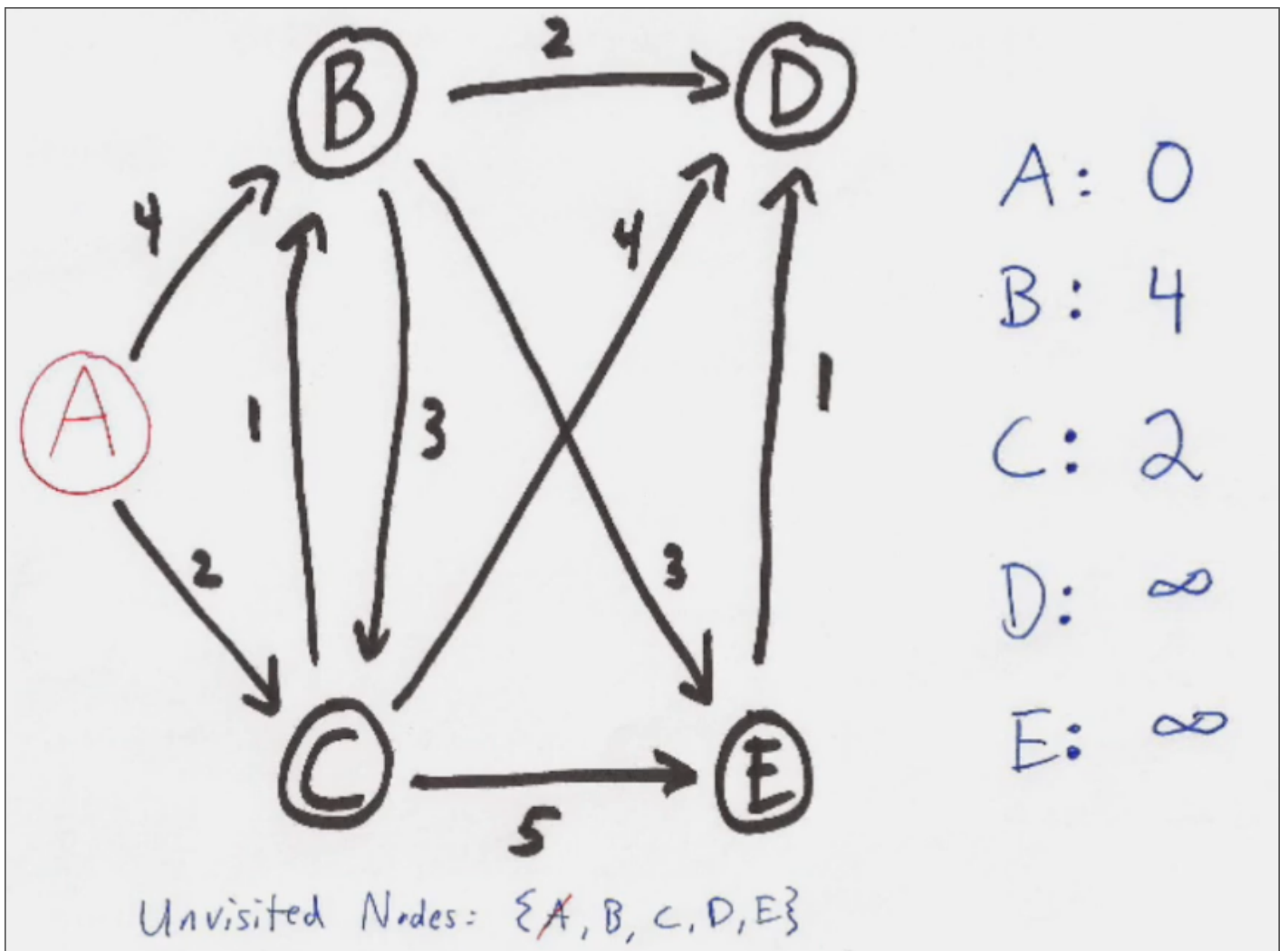
Les images et les explications de cet algorithme viennent de [cette vidéo](#).

Au départ, on indique que le coût pour atteindre le routeur actuel est 0 et ceux pour atteindre les autres routeurs est infinie, car on n'a pas encore calculé le coût.

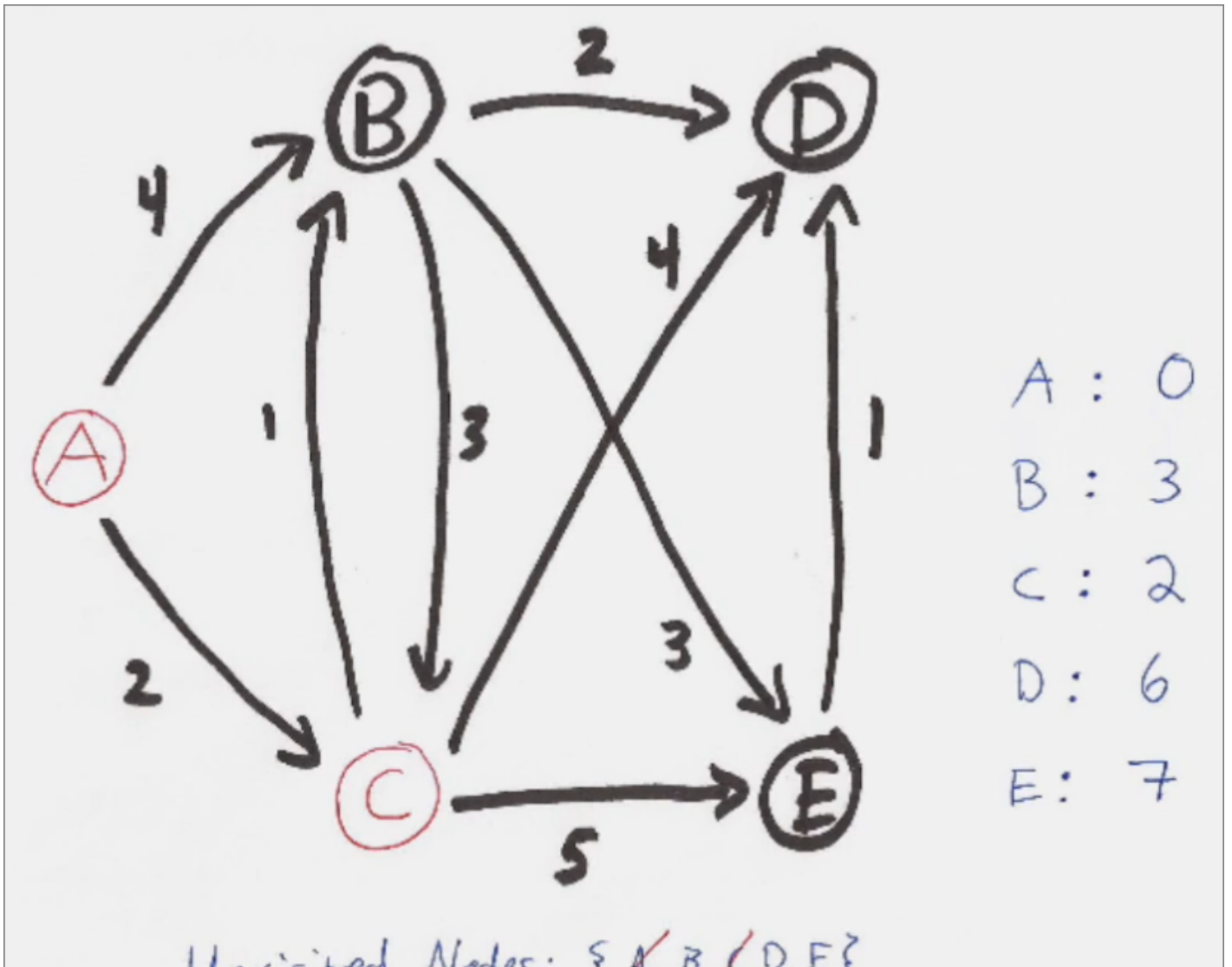
On garde également une liste des routeurs à "visiter".



Ensuite, on peut compléter le tableau des couts en indiquant les couts pour atteindre les routeurs voisins du routeur actuel.

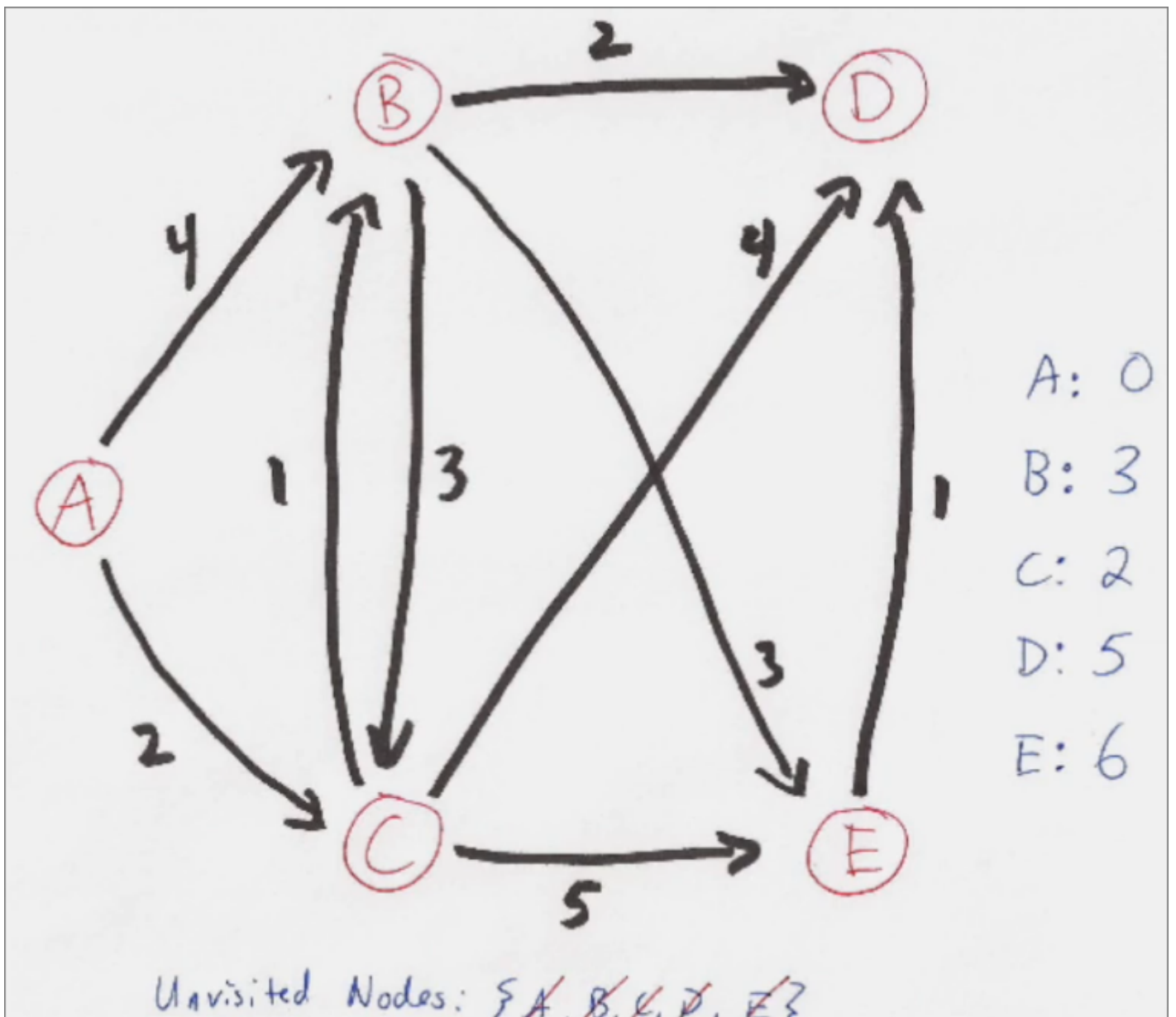


On peut ensuite faire la même chose depuis le routeur ayant le cout le plus bas, dans ce cas-ci, le routeur C.



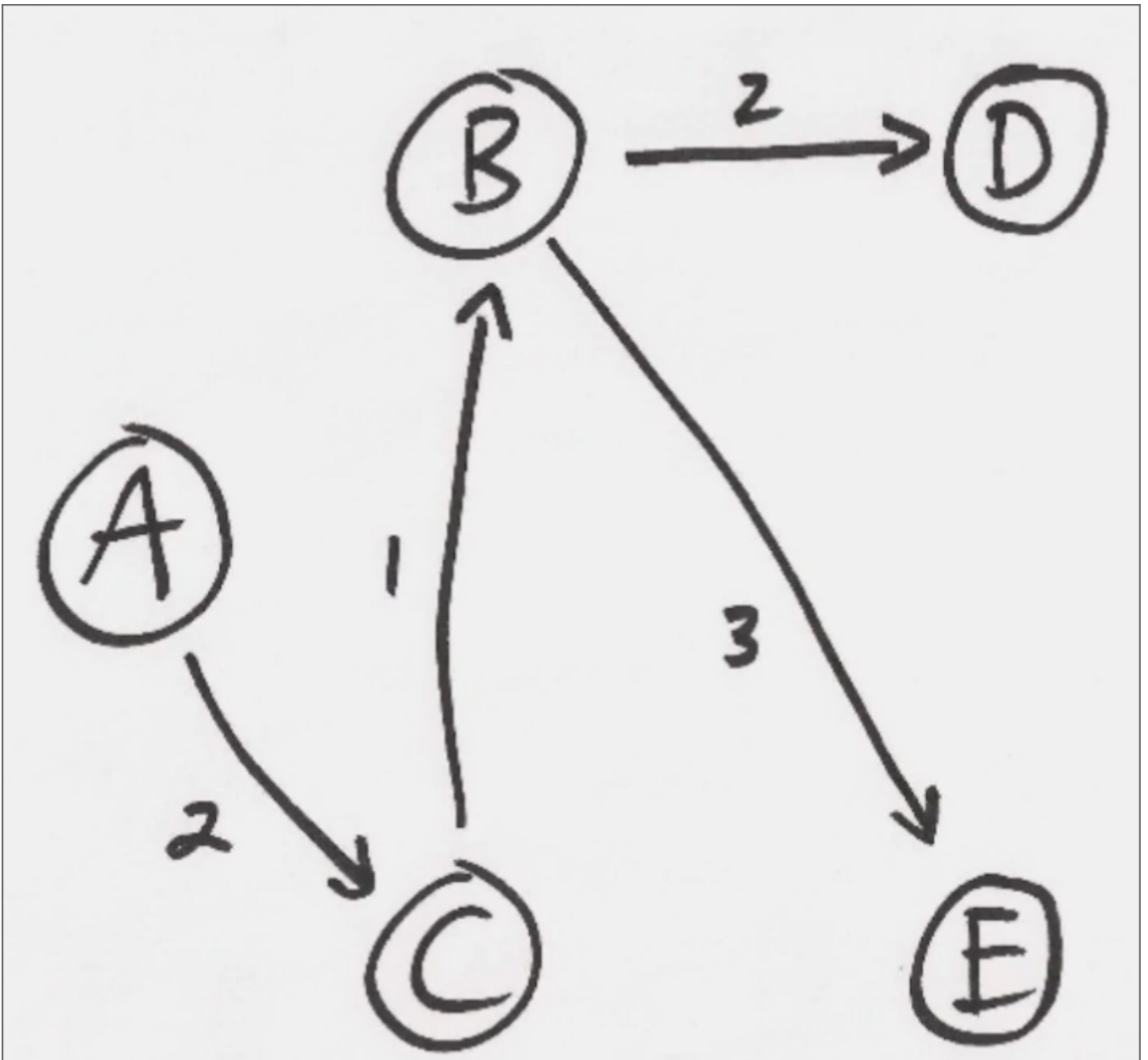
Depuis C, on va ainsi regarder pour chaque routeur voisin de C, quel est le cout pour y accéder. Si le cout total (donc le cout pour atteindre C + le cout pour atteindre le routeur en question) est plus petit que le cout noté précédemment, on met ainsi le cout à jour.

On continue alors le processus jusqu'à avoir fait cette vérification depuis tous les routeurs à visiter.



Une fois cela fait pour tous les routeurs, on peut alors établir quel est le chemin le plus court pour atteindre chaque routeur depuis le routeur courant (A) :





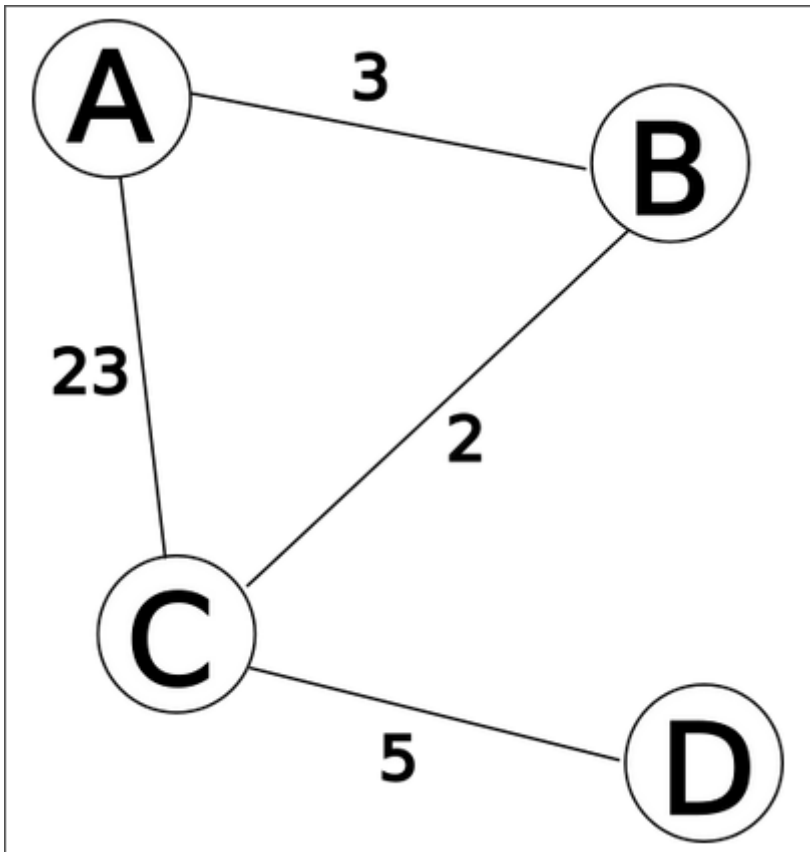
Sur base de cela, nous pouvons donc créer la table de routage. Dans le cas de A, c'est assez simple, car tous les paquets seront envoyés à C.

## Routage dynamique décentralisé

À la place d'avoir une carte complète du réseau, on garde simplement une matrice des distances entre les différents routeurs voisins. Il est ainsi possible de savoir, pour chaque routeur cible, quel routeur doit être utilisé (car plus rapide). Ceci a un grand avantage pour de grands réseaux parce qu'il n'est pas nécessaire de connaître la carte complète du réseau pour pouvoir envoyer des paquets.

### Algorithme vecteur de distance

Imaginons le réseau de routeurs suivant :



Au départ, chaque routeur établit une matrice indiquant le cout pour aller à chacun de leur voisin :

T=0	from	via	via	via	via	from	via	via	via	via	from	via	via	via	via	from	via	via	via	via
	A	A	B	C	D	B	A	B	C	D	C	A	B	C	D	D	A	B	C	D
	to A					to A	3				to A	23				to A				
	to B		3			to B					to B		2			to B				
	to C			23		to C			2		to C					to C			5	
	to D					to D					to D				5	to D				

Ensuite, tous les routeurs envoient ces informations sur le réseau à leur voisin. Ainsi A envoie sa matrice à B et C, B l'envoie à A et C, C l'envoie à A, B et D et D l'envoie à C.

Sur base de cette information, chaque routeur peut recalculer les couts. Ainsi A va recalculer le cout d'aller à C sur base de la matrice de B et va remarquer qu'aller à C en passant par B est beaucoup plus rapide ( $3 + 2 = 5$  contre 23).

Ils vont ainsi mettre à jour leurs matrices :

T=1	from	via	via	via	via	from	via	via	via	via	from	via	via	via	via	from	via	via	via	via
	A	A	B	C	D	B	A	B	C	D	C	A	B	C	D	D	A	B	C	D
	to A					to A	3		25		to A	23	5			to A			28	
	to B		3	25		to B					to B	26	2			to B			7	
	to C		5	23		to C	26		2		to C					to C			5	
	to D			28		to D			7		to D				5	to D				

Ensuite, le processus va se répéter jusqu'à ce que la carte de tout le monde soit complète. Ainsi A va par exemple apprendre le cout pour aller à D est seulement de 10 en passant par B ( $3+7 = 10$ ).

T=3	<b>from</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>from</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>from</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>from</b>	<b>via</b>	<b>via</b>	<b>via</b>	<b>via</b>
	<b>A</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>B</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>C</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>D</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
	<b>to A</b>					<b>to A</b>	3		7		<b>to A</b>	23	5		15	<b>to A</b>			10	
	<b>to B</b>		3	25		<b>to B</b>					<b>to B</b>	26	2		12	<b>to B</b>			7	
	<b>to C</b>		5	23		<b>to C</b>	8		2		<b>to C</b>					<b>to C</b>			5	
	<b>to D</b>		10	28		<b>to D</b>	13		7		<b>to D</b>	33	9		5	<b>to D</b>				

Une fois la table de tout le monde complétée, on peut ainsi définir la table de routage assez simplement en regardant par où il faut passer pour atteindre chaque routeur afin d'avoir la distance la plus courte.

Ainsi, la table de A indique par exemple que tous les paquets seront envoyés à B.

Lorsqu'un ou plusieurs liens avec un routeur est down, les routeurs qui le remarquent indique que le lien vers ces routeurs a un cout infini. Il propage alors cette information dans le réseau pour établir à nouveau la matrice et la table de routage de tous les autres routeurs. Cela se fait de manière assez lente ce qui fait que pendant un certain temps certaines destinations pourrait être inaccessibles.