

# Couche Réseau

## Introduction au routage



## Introduction aux Réseaux

Eric Ramat

[ramat@lisic.univ-littoral.fr](mailto:ramat@lisic.univ-littoral.fr)

Université du Littoral Cote d'Opale  
Licence Informatique 3ème année

# Couche Réseau

## Introduction au routage



## Introduction aux Réseaux

# Modèles OSI et Internet

## Architecture Internet

Applications

Transport

Réseaux

Liaison

Physique

## Données

Message

Segment

Datagramme

Trame

Chaine de bits

## Architecture OSI

Applications

Présentation

Session

Transport

Réseaux

Liaison

Physique

# Plan

## **Généralités**

Services et fonction de la couche réseau  
Modèles de service avec/sans connexion

## **Introduction au routage**

Principe du routage  
Classes d'algorithmes de routage

## **Algorithme par états de liens**

Principe  
Exemple

## **Algorithme à vecteurs de distances**

Principe

## **Autres algorithmes de routage**

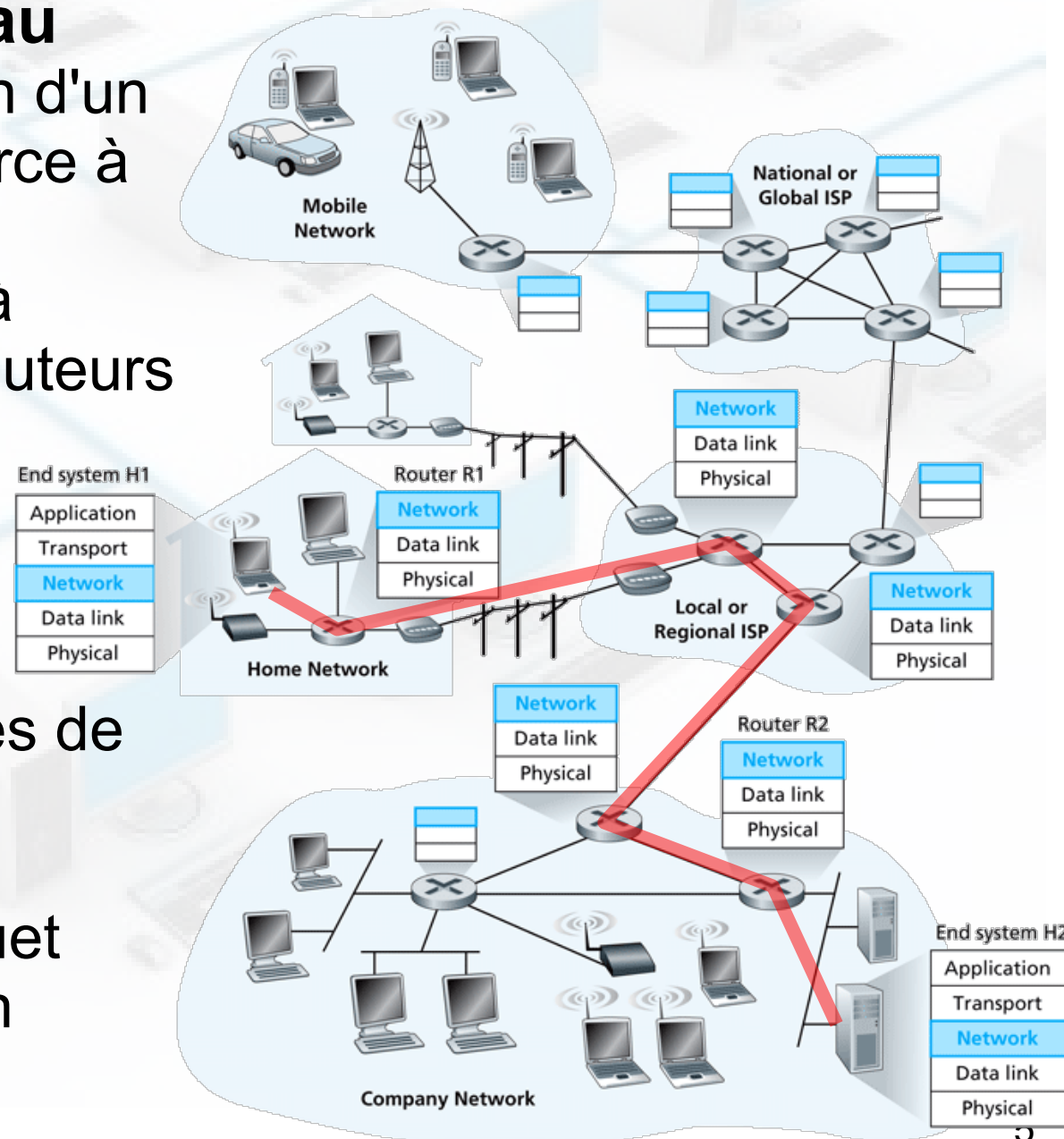
# Introduction

## But de la couche réseau

- Assurer la transmission d'un paquet d'un nœud source à un nœud destination
- Déterminer le chemin à travers les différents routeurs

## Services et fonctions

- Routage : le choix de l'itinéraire à emprunter.  
Fait par des algorithmes de routage
- Réexpédition : la transmission d'un paquet entrant vers une liaison sortante



# Principe

## **Sur le nœud source**

- la couche réseau récupère des messages de la couche transport,
- pour chaque message, elle construit un (ou plusieurs) paquet(s),
- la couche réseau envoie chaque paquet à la couche liaison.

## **Sur chaque nœud intermédiaire (routeur)**

- la couche réseau récupère les paquets de la couche liaison,
- pour chacun d'entre eux, elle construit un nouveau paquet,
- la couche réseau envoie chaque paquet à la couche liaison.

## **Sur le nœud destination,**

- la couche réseau récupère des paquets de la couche liaison,
- elle extrait les données de chaque paquet et les envoie à la couche transport

# Modèles de service

## **Au niveau de la couche réseau deux modes de communication « s'affrontent »**

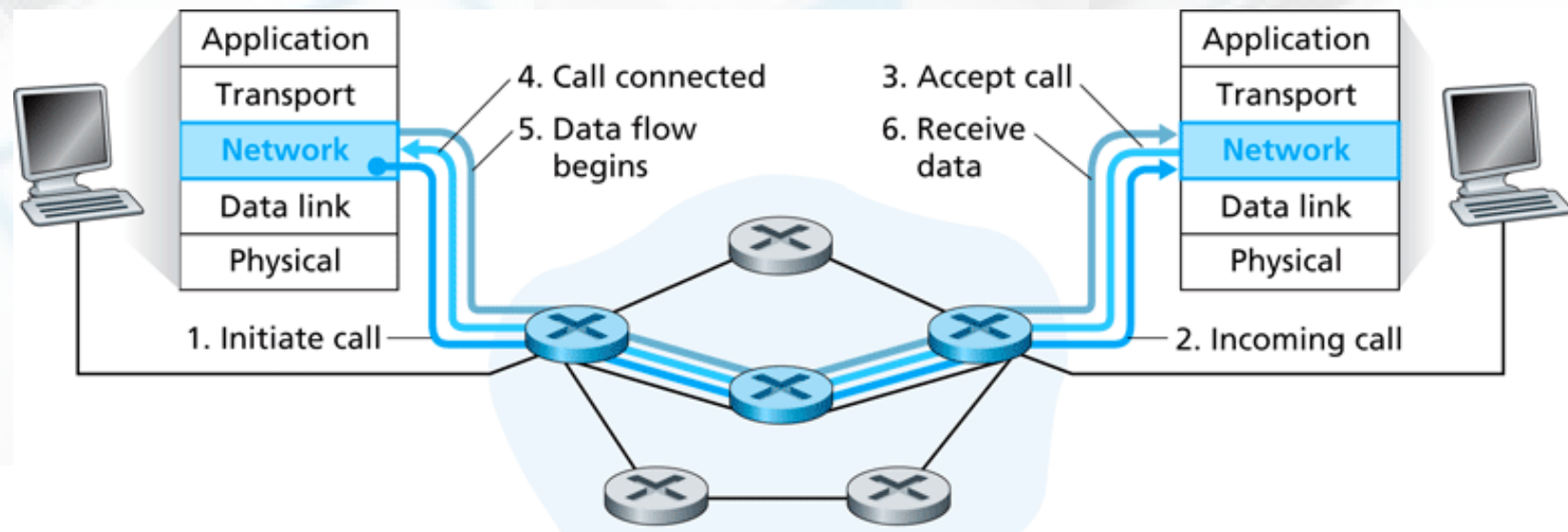
- Le modèle avec connexion
  - Plutôt le choix des opérateurs de réseaux
- Le modèle sans connexion
  - Plutôt le choix de la communauté internet



# Modèle orienté connexion (circuit virtuel)

## Une connexion (circuit virtuel)

- analogie avec les circuits physiques téléphoniques
- doit être établie avant tout envoi entre deux nœuds
- une « route » est calculée à chaque connexion
  - calculer une route au moment de la connexion
  - emprunter cette route pour transférer chaque paquet tant que dure la connexion
- chaque paquet comprend la référence du circuit virtuel
  - les routeurs font des commutations
    - à ne pas confondre avec les switches



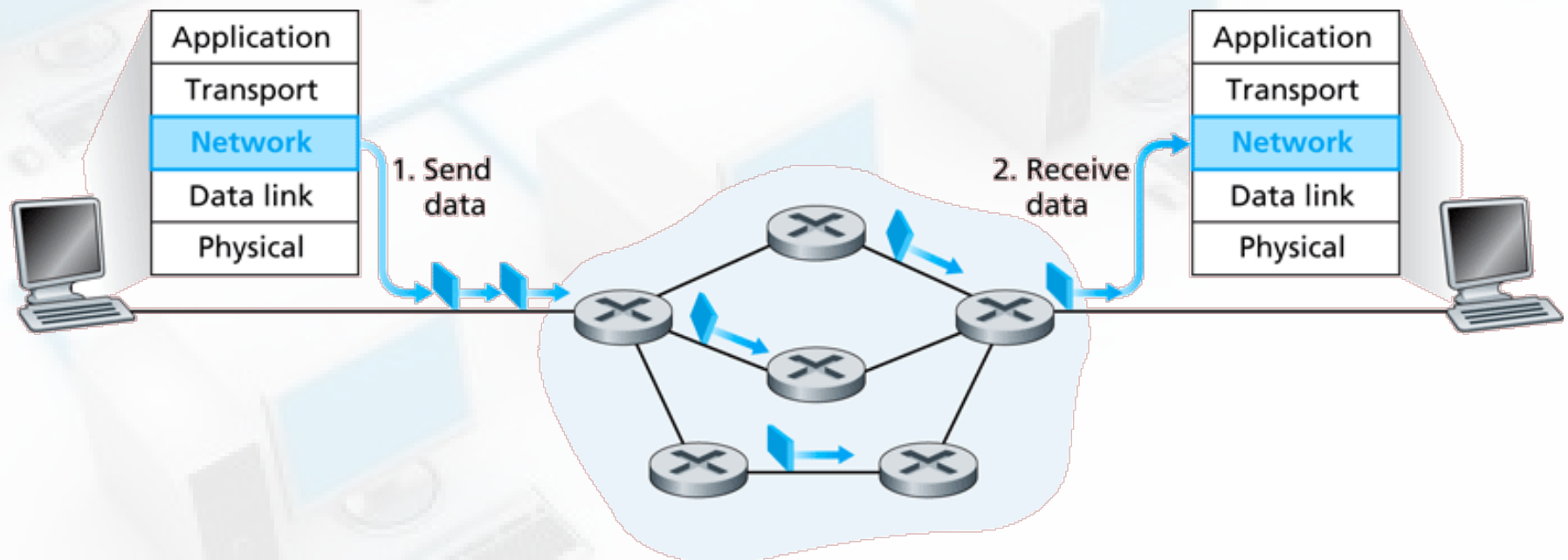


# Modèles sans connexion (datagramme)

## Principe

- les paquets sont transportés de façon indépendante
- sont appelés datagramme (par analogie au télégramme)
- comprend l'adresse de destination
- nécessite un service adressage

## Un chemin est calculé pour chaque paquet



# Avec ou sans connexion ?

## **Commutation : efficacité**

- temps : il n'est pas nécessaire de recalculer une route pour chaque paquet
- espace : une table de commutation à chaque nœud gère les références actives des circuits virtuels. Son encombrement est faible

## **Routage : souplesse**

- chaque paquet peut emprunter un chemin différent
- en cas de congestion ou de panne, cela s'avère particulièrement intéressant

# Plan

## Généralités

Services et fonction de la couche réseau  
Modèles de service avec/sans connexion

## Introduction au routage

Principe du routage

Classes d'algorithmes de routage

## Algorithme par états de liens

Principe

Exemple

## Algorithme à vecteurs de distances

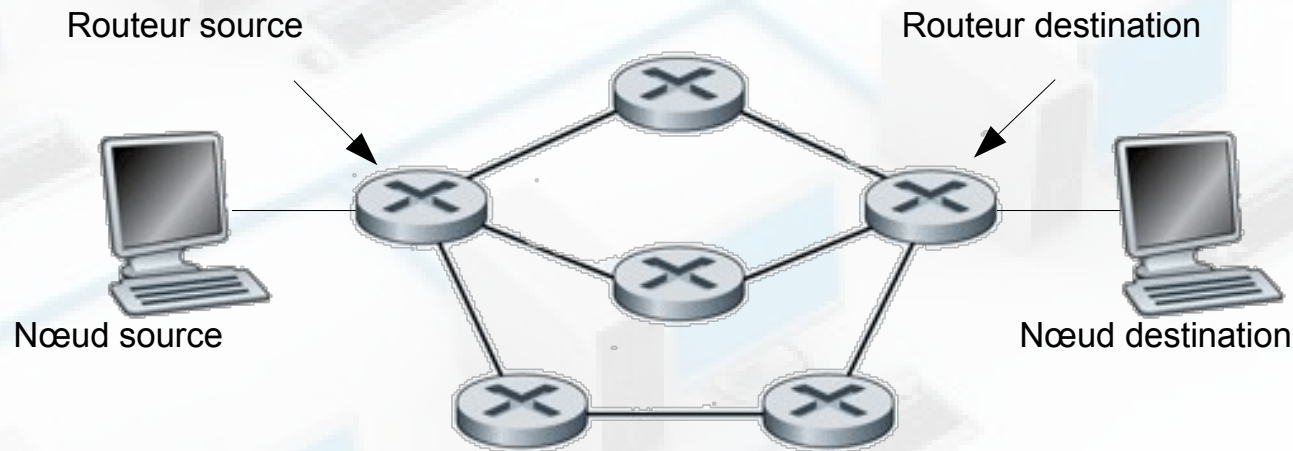
Principe

## Autres algorithmes de routage

# Principe du routage

## Principe

- Le routage est utilisé en mode sans connexion.
- Il consiste à
  - calculer une route pour transférer chaque paquet
- Les équipements permettant le routage s'appellent des routeurs



## Objectif

- Déterminer la route précise pour chaque paquet envoyé
- Une fonction du protocole de routage de la couche réseau

# Routage

## Deux fonctions distinctes

- décider au vue d'informations locales (table de routage) et de l'adresse de destination du paquet à qui envoyer le paquet et sur quel réseau le remettre
- construire la table de routage

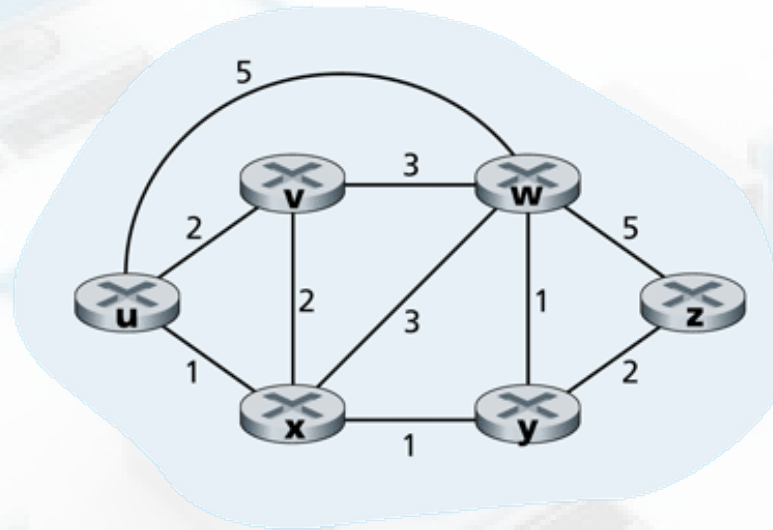
## Algorithme de routage

- principe : en présence d'un groupe de routeurs, reliés par des liens physiques, la mission de l'algorithme de routage consiste à trouver le « bon » parcours entre le routeur source est le routeur destination
  - construire la table de routage
- exécuté sur le routeur

# Quel bon chemin ?

## Critère d'optimisation

- Le chemin le moins onéreux.
- Chaque lien entre les routeurs a un coût qui peut être lié à
  - sa longueur
  - son débit
  - son coût d'utilisation
  - etc ...



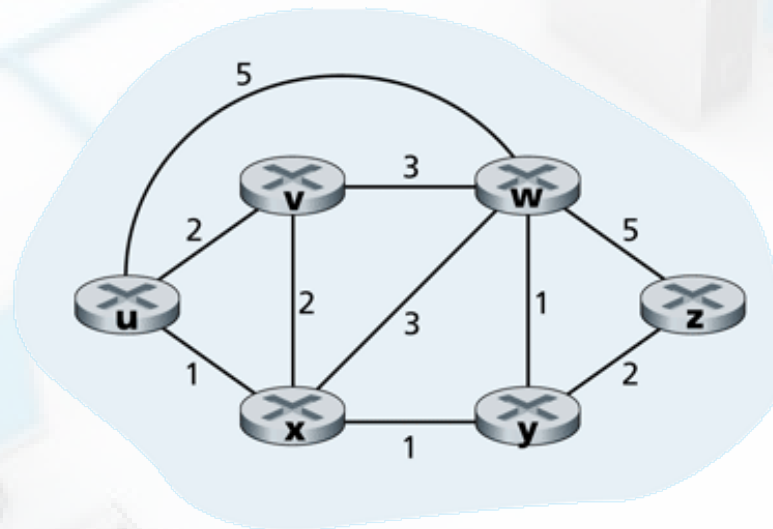
## Exemple

- Le chemin le moins onéreux entre **U** et **W** est **U, X, Y, W**
- Trouvez celui entre **U** et **Z**, comment avez-vous fait ?

# Modèles de représentation

## Objectif

- Choix de la structure de donnée pour appliquer les algorithmes de routage
- Modélisation sous forme de graphe
  - Chaque nœud du graphe est un routeur
  - Chaque lien est une liaison physique
  - A chaque liaison est associé son coût





# Algorithmes de routage

## Principe

- Algorithme de la couche réseau qui a la responsabilité de calculer le chemin qu'un paquet doit suivre
  - Planifier le chemin le moins onéreux de la source à la destination

## Propriétés de algorithmes de routage

- Exactitude
- Simplicité
- Robustesse (capacité d'adaptation aux pannes et changement de topologie)
- Stabilité (convergence vers un état d'équilibre)
- Justice (vis à vis des usagers)
- Optimalité

# Deux classes d'algorithmes de routage

## Algorithmes de routage global

- Calculer en se basant sur une information globale
  - le graphe entier

## Algorithmes de routage décentralisés

- Calculer en se basant sur des information locales
  - pas de connaissance globale du graphe
  - chaque nœud connaît les coûts vers ses voisins auxquels il est directement connecté
  - les nœuds s'échangent les information avec leurs voisins

## Dans les deux cas, l'algorithme peut être

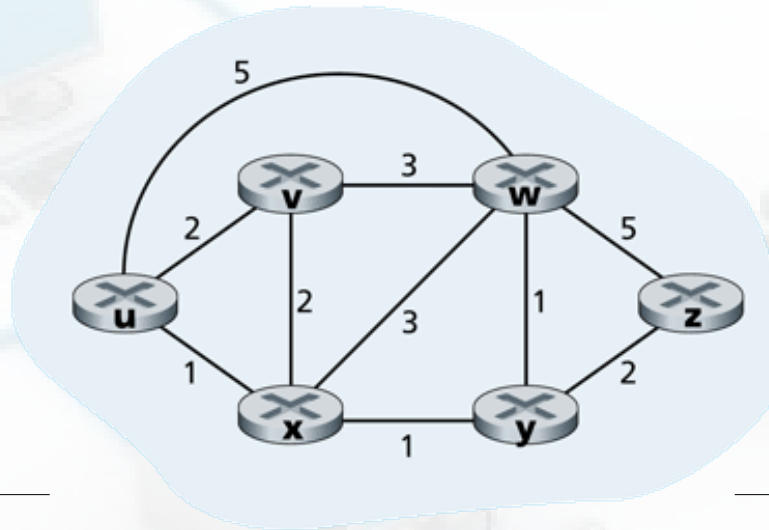
- Statique : les parcours changent très peu, les modifications proviennent souvent d'une intervention humaine
- Dynamique : les parcours s'adaptent automatiquement aux changements de topologie du réseau

# Tables de routage

## Définition

- une table par nœud
- associe à chaque autre nœud du graphe le coût minimal et le lien à suivre
  - à quel routeur voisin doit-on envoyer le datagramme ?

## Exemples



**Trouvez**

**Table de Z**

Destination	Coût	Port
U	4	Y
V	5	Y
W	3	Y
X	3	Y
Y	2	Y

**Table de V**

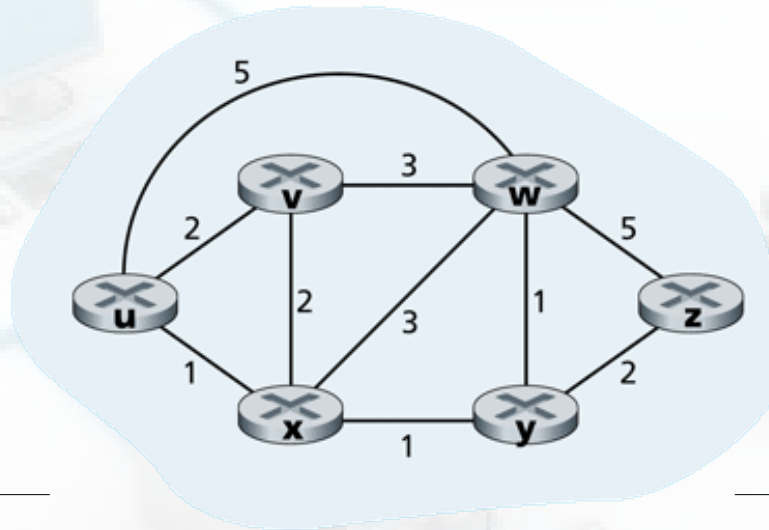
Destination	Coût	Port
U	2	U
W	3	W
X	?	?
Y	?	?
Z	?	?

# Tables de routage

## Définition

- une table par nœud
- associe à chaque autre nœud du graphe le coût minimal et le lien à suivre
  - à quel routeur voisin doit-on envoyer le datagramme ?

## Exemples



**Table de Z**

Destination	Coût	Port
U	4	Y
V	5	Y
W	3	Y
X	3	Y
Y	2	Y

**Table de V**

Destination	Coût	Port
U	2	U
W	3	W
X	2	X
Y	3	X
Z	5	X

# Plan

## Généralités

Services et fonction de la couche réseau  
Modèles de service avec/sans connexion

## Introduction au routage

Principe du routage  
Classes d'algorithmes de routage

## Algorithme par états de liens

Principe  
Exemple

## Algorithme à vecteurs de distances

Principe

## Autres algorithmes de routage

# Algorithme par états de lien (Dijkstra)

## Classe

- Le graphe complet est connu de tous les nœuds
  - tous les nœuds ont la même information
  - accompli avec une diffusion de l'état des liens

## Principe

- Algorithme de « Dijkstra »
- Calculer les chemins les moins coûteux de tous les nœuds à tous les autres
- Pour chaque nœud
  - génère la table de routage du nœud de manière itérative
  - après ***k*** itérations, on connaît le chemin le moins coûteux vers ***k*** destinations

# Algorithme de Dijkstra

## Notations

- $c(i, j)$  : coût du lien entre  $i$  et  $j$
- $D(v)$  : coût courant du chemin de la source à  $v$
- $P(v)$  : nœud précédant  $v$  dans le chemin de la source à  $v$
- $N$  : ensemble des nœuds dont on connaît le coût minimal
- $Adj(i, j)$  : vrai si  $i$  et  $j$  sont adjacents



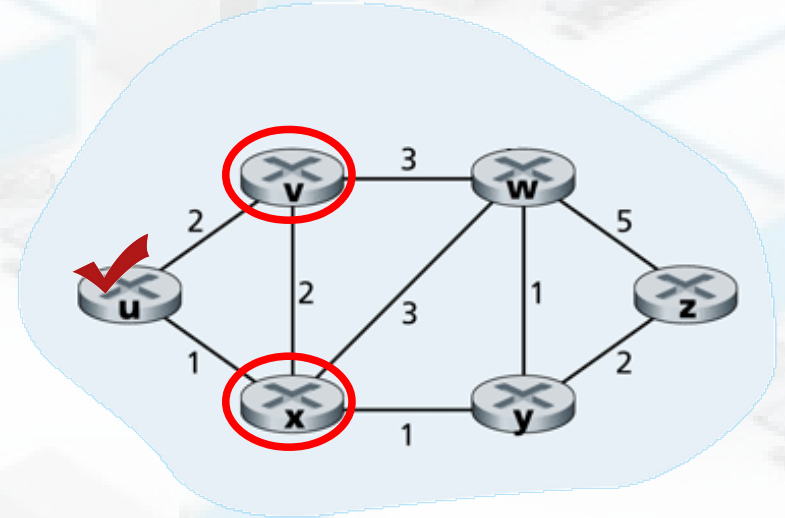
# Algorithme de Dijkstra

```
Dijkstra (entrée: nœud  $x$  , graphe  $G$ ) sortie  $t(x)$   
1  $N = \{x\}$  //  $N$  ensemble de nœuds  
2 Pour tout (nœud  $y \in G$ )  
3   Si  $\text{Adj}(x, y) = \text{vrai}$   
4     Alors  $D(y) = c(x, y)$   
5        $P(y) = x$   
6     Sinon  $D(y) = +\infty$   
7   Fsi  
8 Fpour  
9 Répéter  
10   Trouver  $z \notin N$  tel que  $D(z)$  est minimal  
11    $N = N \cup \{z\}$  //Ajouter  $z$  à  $N$   
12   pour tout (noeud  $y \notin N$  et  $\text{Adj}(w, y) = \text{vrai}$ )  
13      $D(y) = \min( D(y), D(z) + c(z, y) )$   
14      $P(y) = z$   
15   fpour  
15 Jusqu'à ce que tous les nœuds soient dans  $N$ 
```

# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

```
1 N = {x} //N ensemble de nœuds
2 Pour tout (nœud y ∈ G)
3   Si Adj(x,y)= vrai
4     Alors D(y) = c(x,y)
5       P(y) = x
5   Sinon D(y) = +∞
6   Fsi
7 Fpour
8 Répéter
9   Trouver z ∉ N tel que D(z) est minimal
10  N = N ∪ {z} //Ajouter z à N
11  Pour tout (nœud y ∉ N et Adj(z,y)= vrai)
12    D(y) = min( D(y), D(z) + c(z,y) )
13    P(y) = z
14  fpour
15 Jusqu'à ce que tous les nœuds soient dans N
```



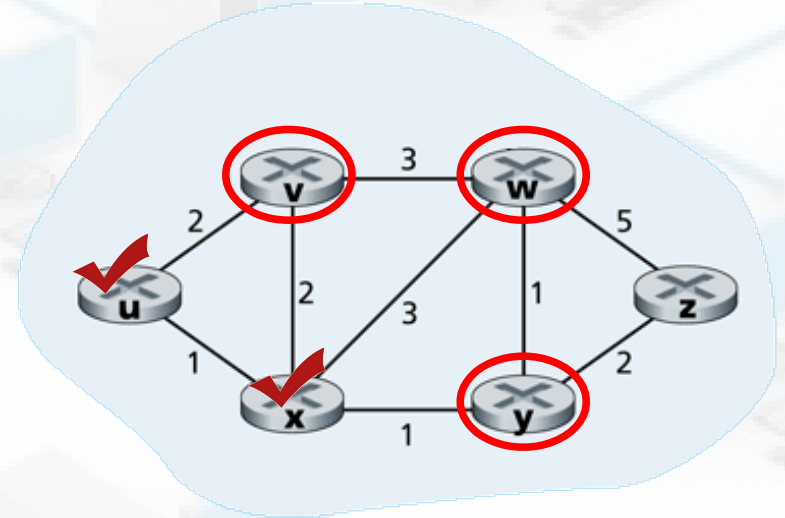
Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	∞, ?	1, u	∞, ?	∞, ?

# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

```

1 N = {x} //N ensemble de nœuds
2 Pour tout (nœud y ∈ G)
3   Si Adj(x,y)= vrai
4     Alors D(y) = c(x,y)
5       P(y) = x
5   Sinon D(y) = +∞
6   Fsi
7 Fpour
8 Répéter
9   Trouver z ∉ N tel que D(z) est minimal
10  N = N ∪ {z} //Ajouter z à N
11  Pour tout (nœud y ∉ N et Adj(z,y)= vrai)
12    D(y) = min( D(y), D(z) + c(z,y) )
13    P(y) = z
14  fpour
15 Jusqu'à ce que tous les nœuds soient dans N
    
```



$$D(v) = \min(2, 1+2)$$

$$D(w) = \min(\infty, 1+3)$$

$$D(y) = \min(\infty, 1+1)$$

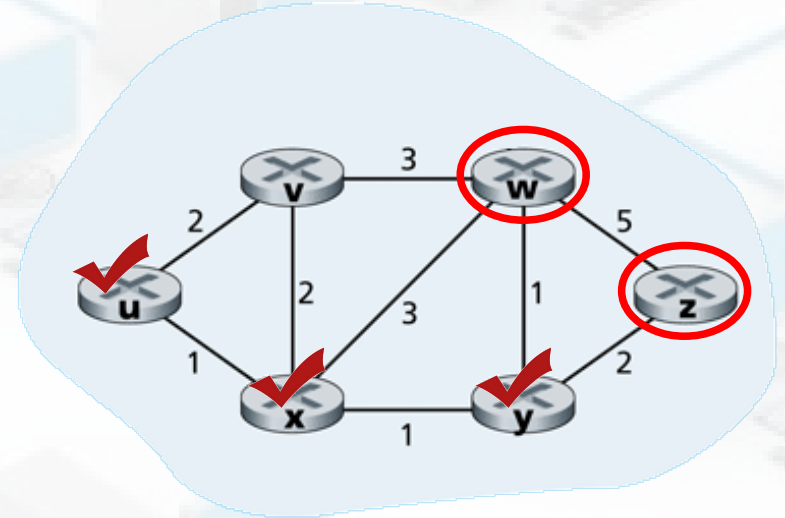
Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	$\infty$ , ?	1, u	$\infty$ , ?	$\infty$ , ?
1	x	{u,x}	v, w, y		4, x		2, x	

# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

```

1 N = {x} //N ensemble de nœuds
2 Pour tout (nœud y ∈ G)
3   Si Adj(x,y)= vrai
4     Alors D(y) = c(x,y)
5       P(y) = x
5   Sinon D(y) = +∞
6   Fsi
7 Fpour
8 Répéter
9   Trouver z ∉ N tel que D(z) est minimal
10  N = N ∪ {z} //Ajouter z à N
11  Pour tout (nœud y ∉ N et Adj(z,y)= vrai)
12    D(y) = min( D(y), D(z) + c(z,y) )
13    P(y) = z
14  fpour
15 Jusqu'à ce que tous les nœuds soient dans N
    
```



$$D(W) = \min(4, 2+1)$$

$$D(z) = \min(\infty, 2+2)$$

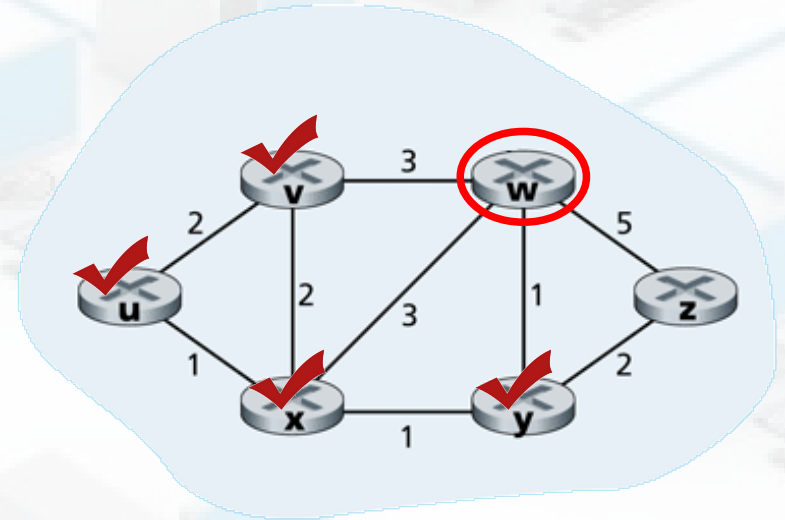
Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	$\infty$ , ?	1, u	$\infty$ , ?	$\infty$ , ?
1	x	{u,x}	v, w, y		<del>4, x</del>		<b>2</b> , x	
2	y	{u,x,y}	w, z		3, y			4, y

# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

```

1 N = {x} //N ensemble de nœuds
2 Pour tout (nœud y ∈ G)
3   Si Adj(x,y)= vrai
4     Alors D(y) = c(x,y)
5       P(y) = x
5   Sinon D(y) = +∞
6   Fsi
7 Fpour
8 Répéter
9   Trouver z ∉ N tel que D(z) est minimal
10  N = N ∪ {z} //Ajouter z à N
11  Pour tout (nœud y ∉ N et Adj(z,y)= vrai)
12    D(y) = min( D(y), D(z) + c(z,y) )
13    P(y) = z
14  fpour
15 Jusqu'à ce que tous les nœuds soient dans N
    
```



$$D(W) = \min(3, 2+3)$$

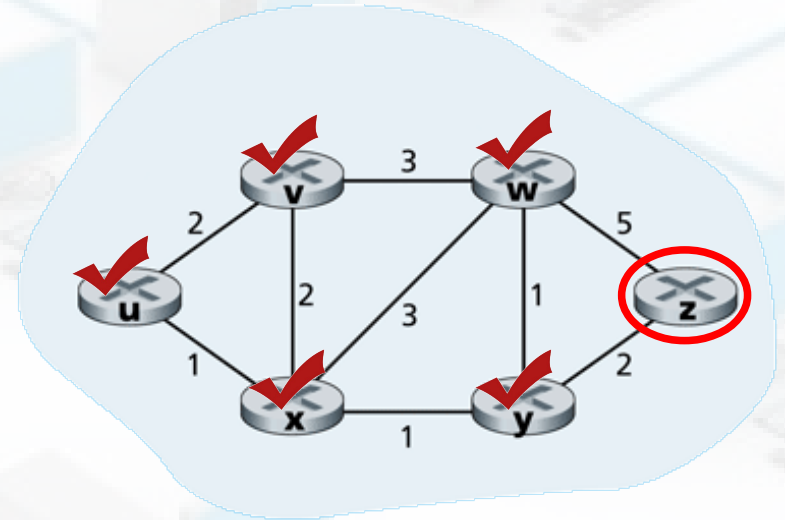
Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	∞, ?	1, u	∞, ?	∞, ?
1	x	{u,x}	v, w, y		4, x		2, x	
2	y	{u,x,y}	w, z		3, y			4, y
3	v	{u,x,y,v}	w					

# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

```

1 N = {x} //N ensemble de nœuds
2 Pour tout (nœud y ∈ G)
3   Si Adj(x,y)= vrai
4     Alors D(y) = c(x,y)
5       P(y) = x
5   Sinon D(y) = +∞
6   Fsi
7 Fpour
8 Répéter
9   Trouver z ∉ N tel que D(z) est minimal
10  N = N ∪ {z} //Ajouter z à N
11  Pour tout (nœud y ∉ N et Adj(z,y)= vrai)
12    D(y) = min( D(y), D(z) + c(z,y) )
13    P(y) = z
14  fpour
15 Jusqu'à ce que tous les nœuds soient dans N
    
```



$$D(z) = \min(4, 3+5)$$

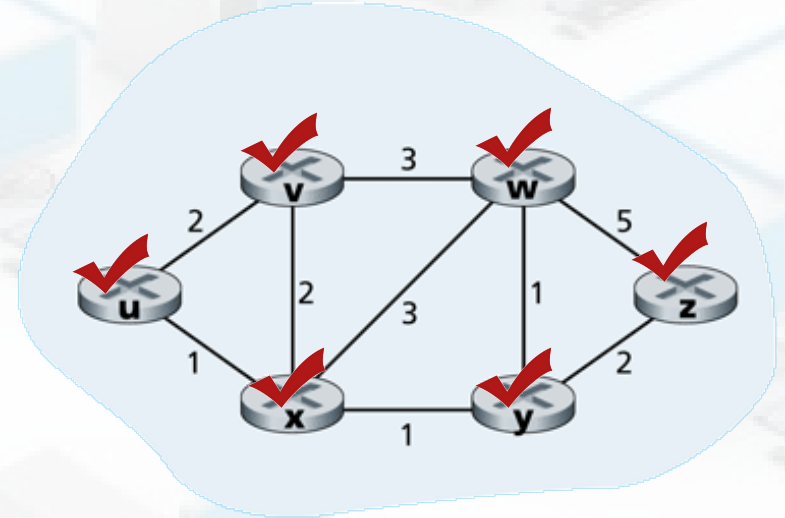
Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	∞, ?	1, u	∞, ?	∞, ?
1	x	{u,x}	v, w, y		<del>4, x</del>		2, x	
2	y	{u,x,y}	w, z		<b>3, y</b>			4, y
3	v	{u,x,y,v}	w					
4	w	{u,x,y,v, w}	z					

# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

```

1 N = {x} //N ensemble de nœuds
2 Pour tout (nœud y ∈ G)
3   Si Adj(x,y)= vrai
4     Alors D(y) = c(x,y)
5       P(y) = x
5   Sinon D(y) = +∞
6   Fsi
7 Fpour
8 Répéter
9   Trouver z ∉ N tel que D(z) est minimal
10  N = N ∪ {z} //Ajouter z à N
11  Pour tout (nœud y ∉ N et Adj(z,y)= vrai)
12    D(y) = min( D(y), D(z) + c(z,y) )
13    P(y) = z
14  fpour
15 Jusqu'à ce que tous les nœuds soient dans N
    
```



Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	∞, ?	1, u	∞, ?	∞, ?
1	x	{u,x}	v, w, y		<del>4, x</del>		2, x	
2	y	{u,x,y}	w, z		3, y			4, y
3	v	{u,x,y,v}	w					
4	w	{u,x,y,v, w}	z					
5	z	{u,x,y,v, w,z}						



# Exemple de l'algorithme de Dijkstra

## Calcul de la table de routage du nœud u

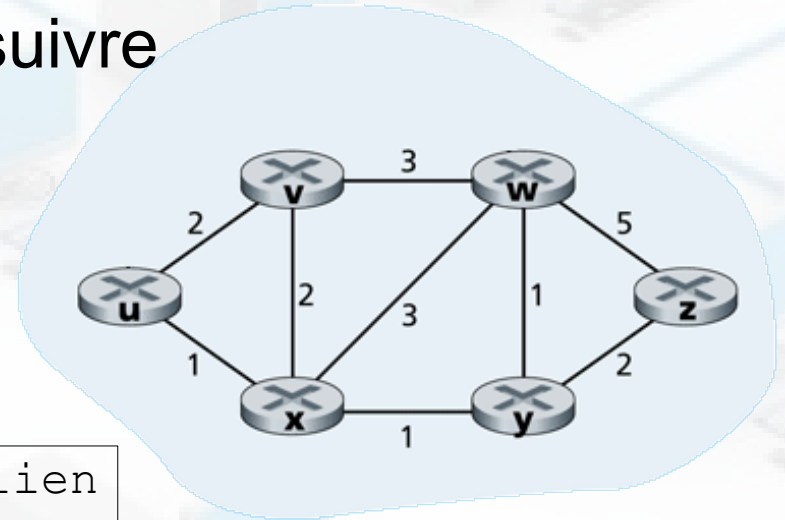
- Pour déduire la table, il suffit de suivre les  $P^*$  à partir de la destination jusqu'à la source **u**,

– Ex. pour **z** :

$P(z)=y \rightarrow P(y)=x \rightarrow P(x)=u$

- Le chemin est u,x,y,z
- Le lien est x
- Le coût est  $D(z)$

dest	coût	lien
V	2	v
W	3	x
X	1	x
Y	2	x
Z	4	x



Étape	z	N	y	D(v), P(v)	D(w), P(w)	D(x), P(x)	D(y), P(y)	D(z), P(z)
Init	-	{u}	-	2, u	$\infty$ , ?	1, u	$\infty$ , ?	$\infty$ , ?
1	x	{u,x}	v, w, y		<del>4, x</del>		2, x	
2	y	{u,x,y}	w, z		3, y			4, y
3	v	{u,x,y,v}	w					
4	w	{u,x,y,v, w}	z					
5	z	{u,x,y,v, w,z}						

# Plan

## Généralités

Services et fonction de la couche réseau  
Modèles de service avec/sans connexion

## Introduction au routage

Principe du routage  
Classes d'algorithmes de routage

## Algorithme par états de liens

Principe  
Exemple

## Algorithme à vecteurs de distances

Principe

## Autres algorithmes de routage

# Algorithme à vecteur distance

## Classe

- chaque routeur dispose d'une information locale
- précisant pour chaque destination connue le meilleur chemin

## Principe

- les informations partielles sont échangées régulièrement entre les routeurs afin de mettre à jour leur connaissances
- échange de vecteur de distance

## Vecteur distance

- associe à chaque destination connue son coût estimé
- équivalent à une table de routage sans les liens

# Algorithme RIP

## *Routing information protocol*

### **Échange d'information (diffusion)**

- périodiquement (toutes les 30s) les routeurs envoient leur vecteur distance à leurs voisins directs
  - vecteur de (destination, coût)

### **A la réception**

- chaque fois qu'un routeur reçoit un VD, il exécute l'algorithme RIP pour éventuellement mettre à jour sa table de routage

# Algorithme RIP

Le vecteur  $v$  a été reçu du routeur  $R$

```
RIP (in:vecteur  $v$ , in:routeur  $R$ , in/out:table  $T$  )
1  pour chaque ligne  $(d, c)$  de  $v$ 
2      si  $d \notin T$  alors
3          ajouter à  $T$   $(d, c+1, R)$ 
4      sinon //déjà dans  $T$  comme  $(d, c_{\text{actuel}}, l_{\text{actuel}})$ 
5          si  $R = l_{\text{actuel}}$  alors
6              mettre à jour  $T$  avec  $(d, c+1, R)$ 
7          sinon //  $R \neq l_{\text{actuel}}$ 
8              si  $c+1 < c_{\text{actuel}}$  alors
9                  remplacer dans  $T$  avec  $(d, c+1, R)$ 
10             finsi
11         finsi
12     finsi
13 fpour
```

# Remarque sur l'algorithme RIP

## ***Nœuds terminaux***

- seule la partie réception est effectuée
  - les stations de travail ne diffusent pas ; seules les routeurs le font

## **A l'initialisation des routeurs**

- les tables de routage sont initialisées avec l'ensemble des adresses des réseaux auxquels le routeur est directement connecté
- le coût minimum (égal à 1) est alors associé à ces adresses destinations

## **Ne tient pas compte du coût réel**

- utilise le nombre de sauts (coût de 1)

# Plan

## **Généralités**

Services et fonction de la couche réseau  
Modèles de service avec/sans connexion

## **Introduction au routage**

Principe du routage  
Classes d'algorithmes de routage

## **Algorithme par états de liens**

Principe  
Exemple

## **Algorithme à vecteurs de distances**

Principe

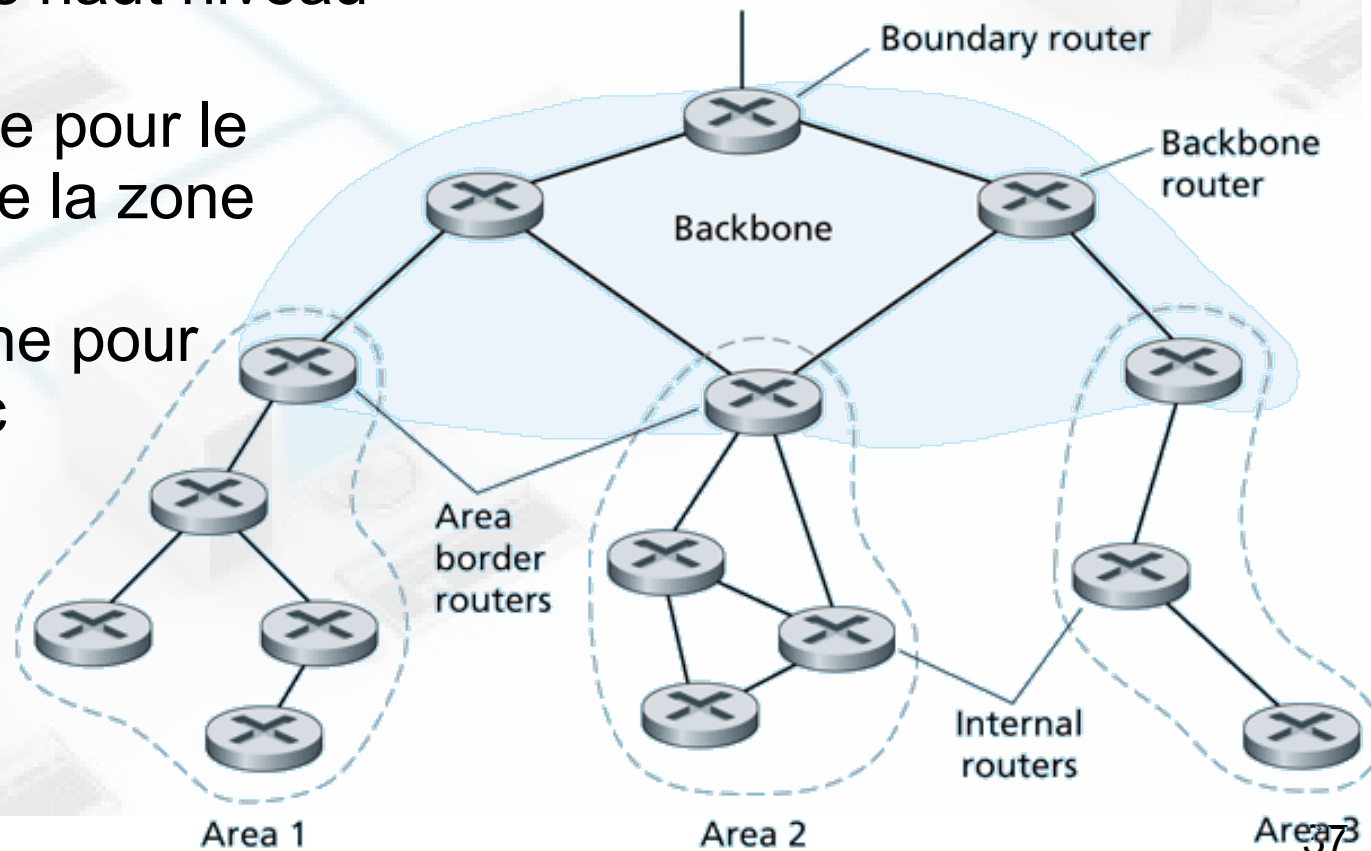
## **Autres algorithmes de routage**



# Routage hiérarchique

## Internet contient plusieurs milliers de routeurs

- Échanger et mettre à jour les table de routage devient impossible
- On a recourt à un routage hiérarchique
  - Avec des « **zones autonomes** » reliées par des FAI de plus haut niveau
  - Routage interne pour le trafic au sein de la zone
  - Routage externe pour tout autre trafic



# Autres algorithmes de routage

## Autres classes d'algorithmes de routage

- algorithmes d'états de liens optimisés
  - utilisé dans les réseaux mobile ad-hoc
- algorithmes à vecteurs de chemins

## OSPF (*Open Shortest Path First*)

- une instance de Disjkstra

## BGP (*Border Gateway Protocol*)

- algorithme à vecteur de chemin
- pour le routage entre les zones autonomes

**Et bien d'autres ...**

# Résumé

## **But de la couche réseau**

- Acheminer un paquet jusqu'à la destination

## **Le modèle sans connexion nécessite**

- Un mécanisme d'adressage
- Un mécanisme de routage

## **Le routage = deux fonctions**

- Le calcul des tables de routage
- L'acheminement des datagrammes au routeur suivant

## **Algorithme de routage**

- Global / décentralisé / dynamique / adaptatif
- Ces algorithmes doivent être simples et efficaces