
Le niveau paquet

LA COUCHE RÉSEAU

—

LE ROUTAGE

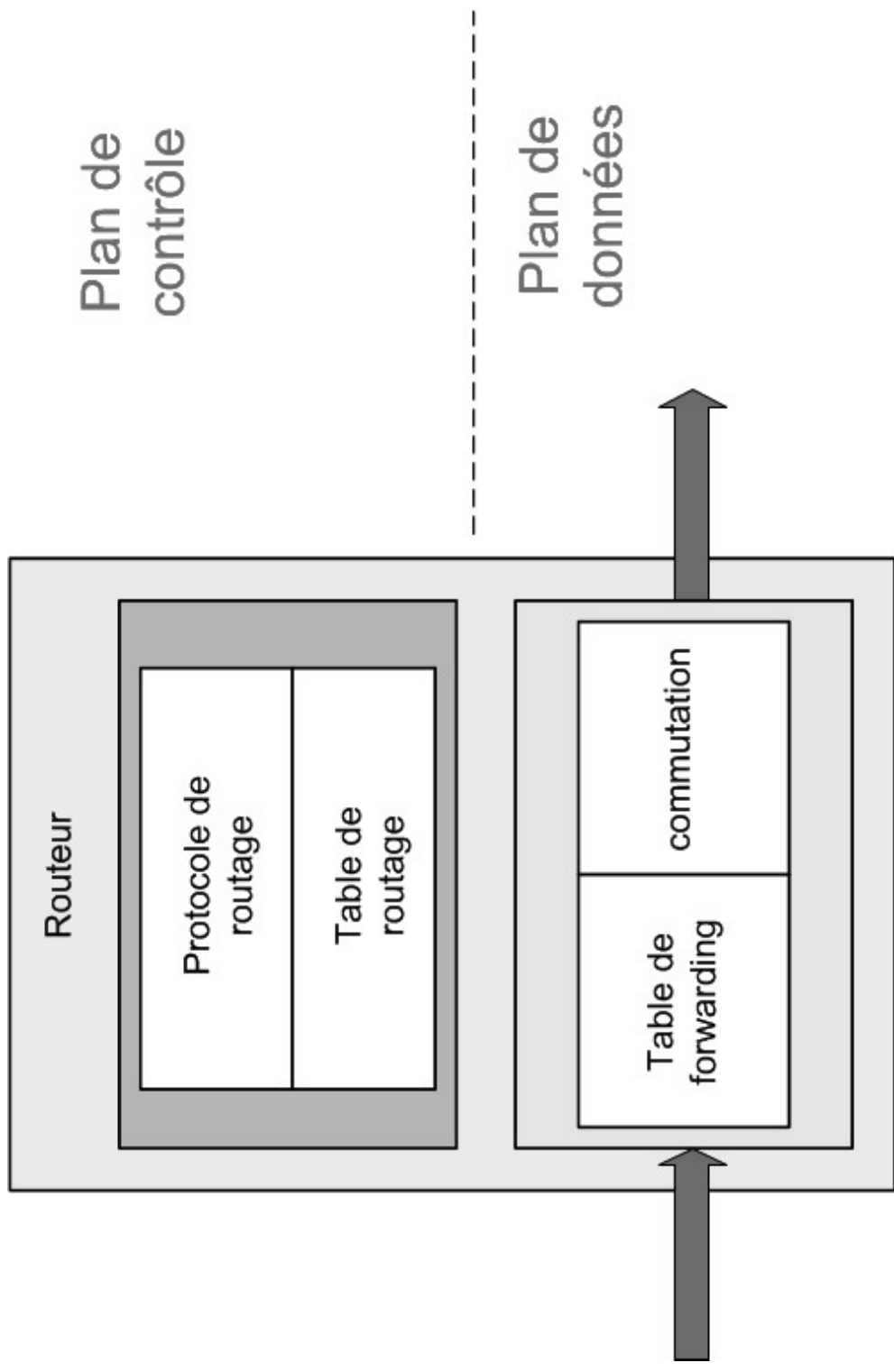
Introduction

- IP repose sur un routage de proche en proche
 - On cherche à chaque fois le prochain saut à effectuer
- Que ce soit pour IPv4 ou pour IPv6, les routeurs reposent toujours sur des tables pour décider du chemin que doit emprunter un paquet
 - Table de routage
 - Table de forwarding
 - Table de correspondance d'adresses
- Attention: **IP n'est pas un protocole de routage !**
 - C'est un protocole routable, il explique comment le réseau est organisé
 - Il ne fournit pas de chemins mais un moyen de les mettre en œuvre

Introduction

- Un **protocole de routage** va expliquer comment est réalisé l'acheminement des paquets dans un réseau
- Pour le fonctionnement d'un routeur, on distingue donc deux niveaux
 - **Le plan de contrôle**
 - Définit la signalisation et les informations à échanger pour rendre un service
 - Mise en œuvre du protocole de routage
 - **Le plan de données**
 - Assure le bon traitement des paquets en fonctions des instructions définies par le plan de contrôle
 - Assure les fonctions de niveau 2

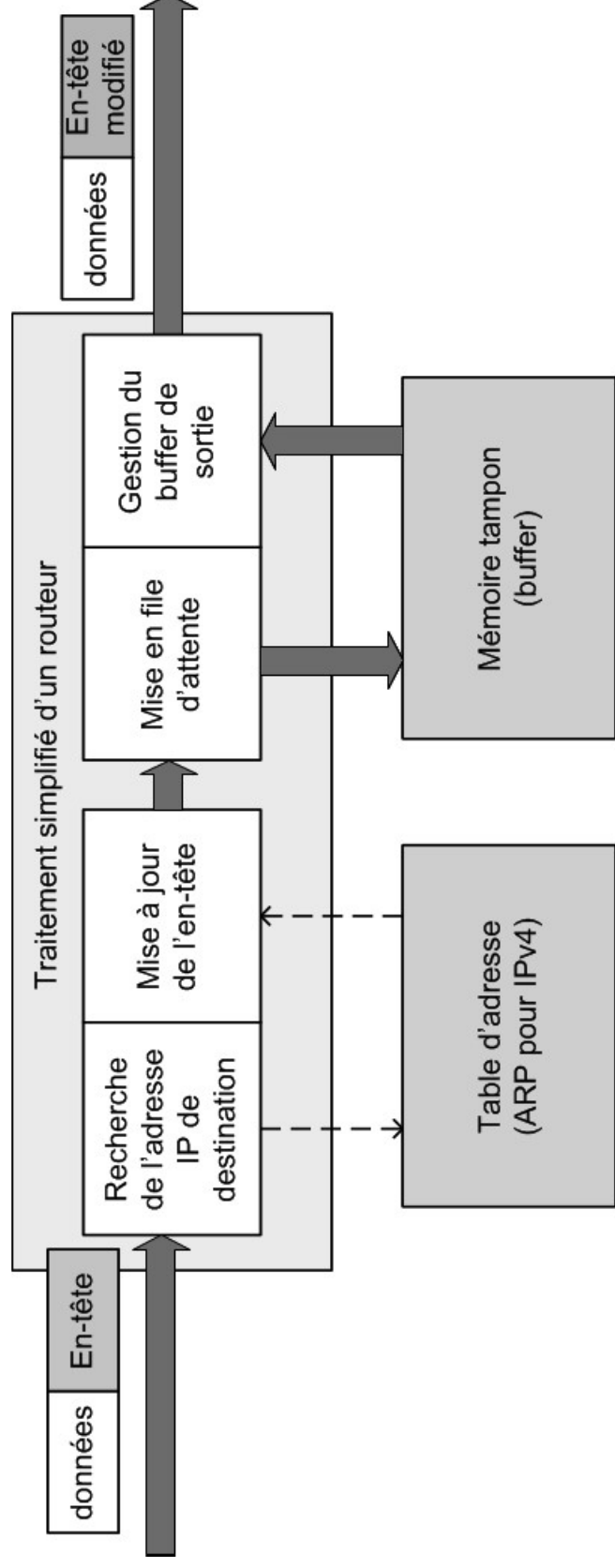
Architecture générique d'un routeur



- En pratique les tables de routage et de forwarding sont souvent fusionnées

Architecture générique d'un routeur

- Traitement simplifié d'un routeur
 - Les deux premiers traitements sont effectués pour chaque interface d'entrée
 - La mise en file d'attente et la gestion du buffer de sortie pour chaque interface de sortie



Architecture générique d'un routeur

- Concrètement la mise-à-jour de l'en-tête consiste à modifier
 - Les informations de niveau 2 et en particulier les @MAC
 - L'adresse du nœud émetteur est remplacée par l'@MAC de l'interface du routeur sur laquelle le paquet sera envoyé
 - L'adresse du destinataire est positionnée à l'adresse de prochain saut
 - Mettre à jour le nombre de sauts restants dans l'en-tête IP
- Quelques questions restent encore en suspend néanmoins
 - Comment sont construites ces tables ?
 - Comment gère-t-on les changements dans le réseau ?
 - Quid des boucles ?
 - Comment assurer qu'un chemin est optimal ?
 - Qu'est ce qu'un chemin optimal ?

Les méthodes de routage

- Il existe deux solutions pour établir et maintenir les tables de routage
 - **Routage statique**
 - Les routes sont construites manuellement
 - Pbm: passage à l'échelle impossible
 - ★ *Lorsque le nombre d'éléments constitutifs du réseau est très important*
 - ★ *Lorsque le réseau évolue très fréquemment*
 - On limite souvent le routage statique aux nœuds d'extrémité du réseau
 - **Routage dynamique**
 - Nécessite un protocole de routage dédié
 - ★ *Mise en place de signalisation pour assurer l'échange des tables de routage*
 - Recommandé pour les éléments d'interconnexion comme les routeurs

LES ALGORITHMES DE ROUTAGE

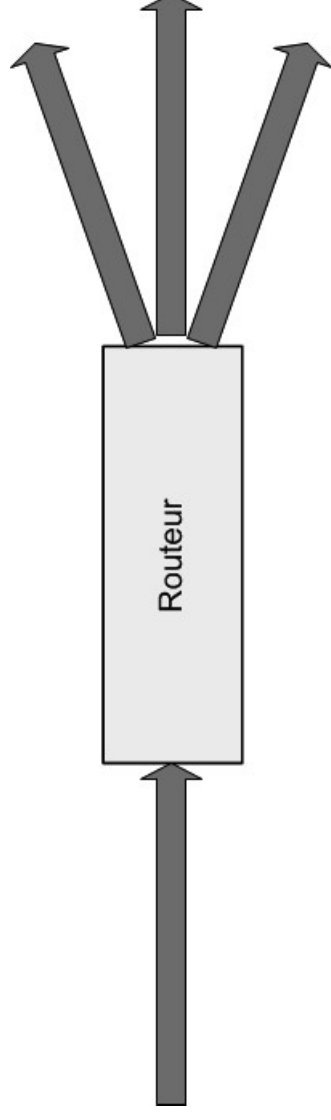
Objectifs

- L'objectif est de partager les tables de routage entre tous les nœuds pour assurer une cohérence d'ensemble
 - Un protocole de routage doit donc permettre l'adaptation des routeurs à l'évolution d'un réseau
 - Apparition et disparition de liens ou de routeurs au cours du temps
 - Pour cela, il doit assurer
 - La sélection d'un chemin de manière optimale
 - L'élimination des boucles de routage
 - Une faible consommation de bande passante et de CPU pour sa signalisation
 - Une convergence rapide vers un résultat (principe de stabilité)
 - Une configuration simple
-

L'inondation

- Principe: Les routeurs retransmettent les paquets sur toutes leurs interfaces (sauf l'interface d'entrée)
 - Solution la plus simple pour assurer la diffusion d'un paquet
 - Pas besoin de table ni de connaître la topologie du réseau
 - Par contre, les routeurs reçoivent plusieurs fois le même paquet et doivent traiter une masse très importante d'information
 - Problème de boucles

- Ne permet pas de passer à l'échelle



Les grandes classes d'algorithmes

- Il existe deux grandes catégories d'algorithmes de routage
 - **Par vecteur de distance**
 - Echange périodique d'informations entre routeurs voisins
 - Traitement effectué de manière indépendante par chaque nœud
 - Chaque routeur à une vision parcellaire du réseau d'ensemble
 - ★ *Il ne connaît que ses voisins et le poids des liens qui les relient*
 - ★ *Il obtient la carte complète du réseau par l'échange de tables entre routeurs adjacents*
 - **Par état de liens**
 - Chaque routeur dialogue avec tous les autres routeurs du réseau
 - Il détermine localement les plus courts chemins de tous les nœuds vers lui-même
 - ★ *Plus de problèmes de boucles*
 - Nécessite une connaissance globale du réseau (carte)

Les algorithmes par vecteur de distance

- Ces algorithmes vont utiliser deux informations
 - La direction
 - La distance
 - Plusieurs métriques sont envisageables pour déterminer la distance
 - Le délais d'acheminement
 - Le nombre de sauts
 - Le nombre de paquets en attente sur un chemin
 - ...
 - La table de routage va se composer de trois informations
 - La destination à atteindre
 - Le prochain saut à emprunter
 - La distance qui nous sépare de la destination
 - Chaque noeud échange les entrées de la table avec ses voisins
 - Echange d'informations de manière périodique et lorsqu'un changement survient
-

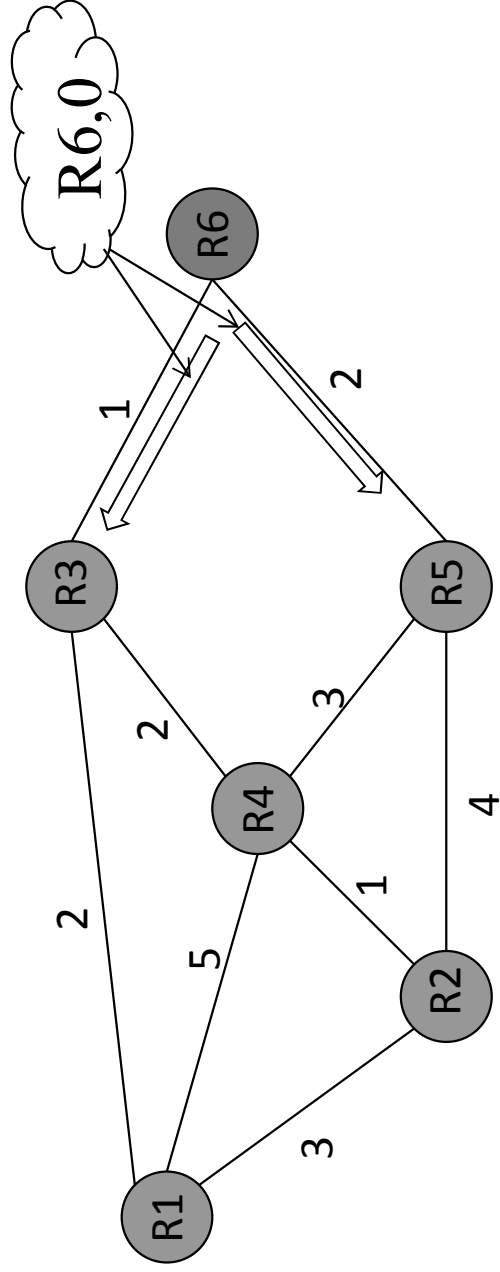
L'algorithme de Bellman-Ford

- La technique du routage par vecteur de distance est également appelée l'algorithme de Bellman-Ford en hommage à ses concepteurs
 - **Principe de l'algorithme** pour une destination D
 - Prérequis: Un routeur doit pouvoir évaluer la distance qui le sépare de ses voisins
 - Si la distance est le nombre de sauts alors la distance vaut 1, pour le délais il peut utiliser ICMP etc.
 - Au début de l'algorithme la distance de tous les nœuds vers D vaut ∞
 - Sauf bien sur la distance de D vers D qui vaut 0
 - D envoie son vecteur de distance à ses voisins toutes les T secondes
 - Lorsqu'un voisin reçoit un vecteur, il met à jours les informations de sa table en cherchant la distance minimum
 - Il envoie ensuite l'information à ses voisins (on reprend à l'étape 3)
 - Pour l'ensemble des destinations possible on applique le même principe en cherchant à chaque fois le voisin avec la plus faible distance pour aller à D
-

L'algorithme de Bellman-Ford

- Exemple d'application pour la destination R6

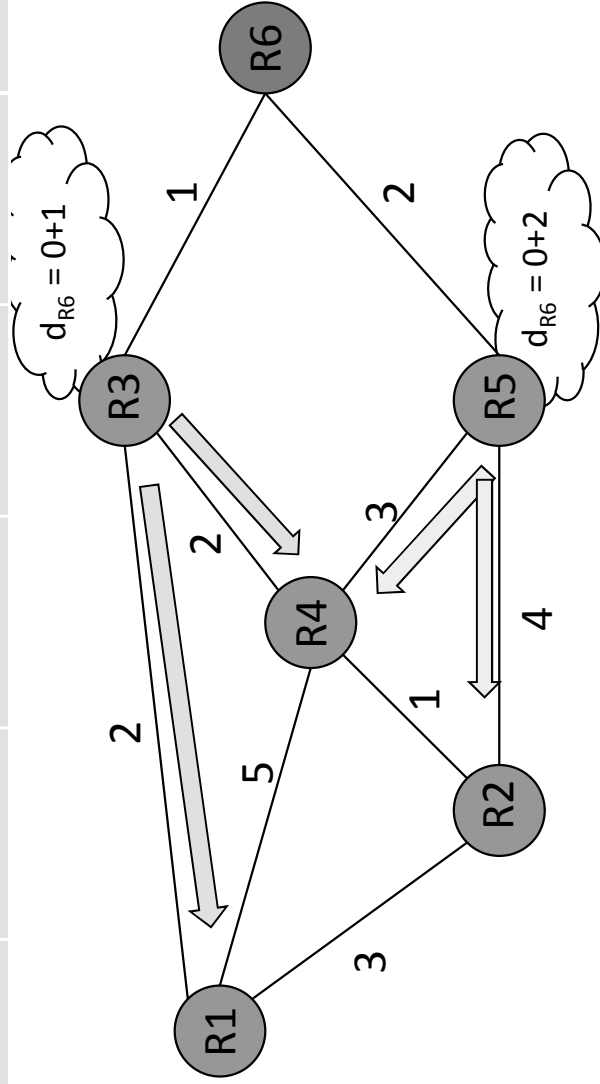
| Itération | R1 | R2 | R3 | R4 | R5 | R6 |
|-----------|----|----|----|----|----|-------|
| Début | | | | | | R6, 0 |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |



L'algorithme de Bellman-Ford

- Exemple d'application pour la destination R6

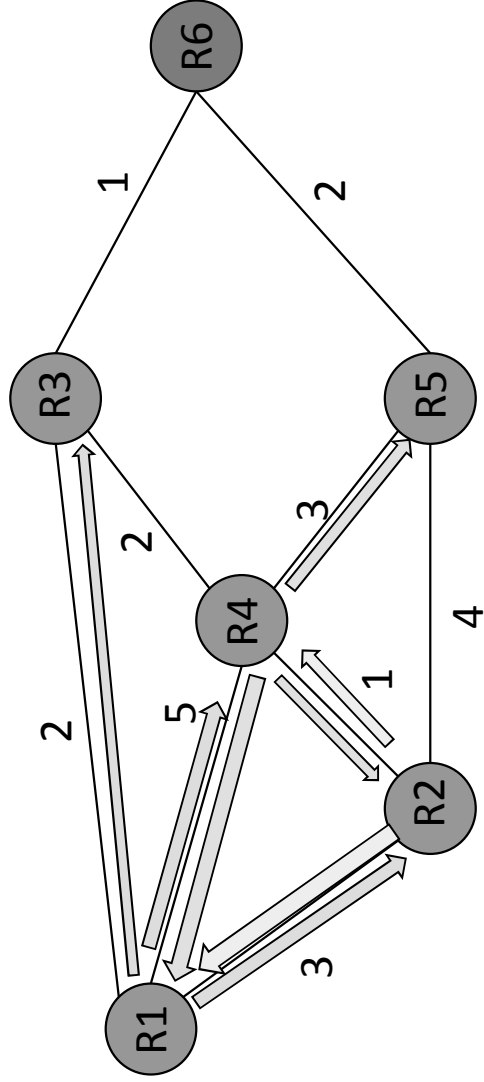
| Itération | R1 | R2 | R3 | R4 | R5 | R6 |
|-----------|----|----|-------|----|-------|-------|
| Début | | | | | | R6, 0 |
| 1 | | | R6, 1 | | R6, 2 | |
| 2 | | | | | | |
| 3 | | | | | | |



L'algorithme de Bellman-Ford

- Exemple d'application pour la destination R6

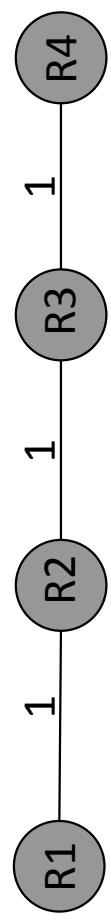
| Itération | R1 | R2 | R3 | R4 | R5 | R6 |
|-----------|-------|-------|-------|-------|-------|-------|
| Début | | | | | | R6, 0 |
| 1 | | | R6, 1 | | R6, 2 | |
| 2 | R3, 3 | R5, 6 | R6, 1 | R3, 3 | R6, 2 | |
| 3 | R3, 3 | R4, 4 | R6, 1 | R3, 3 | R6, 2 | |



L'algorithme de Bellman-Ford

- Le problème de la valeur infinie
 - En théorie l'algorithme à vecteur de distance fonctionne parfaitement mais dans la pratique certaines situations peuvent poser problème
 - Un des inconvénients majeur est qu'il réagit très vite aux bonnes nouvelles et plus lentement aux mauvaises

➤ Exemple



- Table vers R1

| Itération | R1 | R2 | R3 | R4 |
|-----------|-------|-------|-------|-------|
| Début | R1, 0 | R1, 1 | R2, 2 | R3, 3 |
| 1 | | R3, 3 | R2, 2 | R3, 3 |
| 2 | | R3, 3 | R2,4 | R3, 3 |
| 3 | | R3,5 | | R3,5 |

Rupture du lien R1-R2 → 1

- Le processus est infini

L'algorithme de Bellman-Ford

- Technique de l'horizon éclaté (split horizon)
 - Un routeur n'annonce jamais la route à son voisin v si celui-ci est le prochain saut à contacter pour cette destination

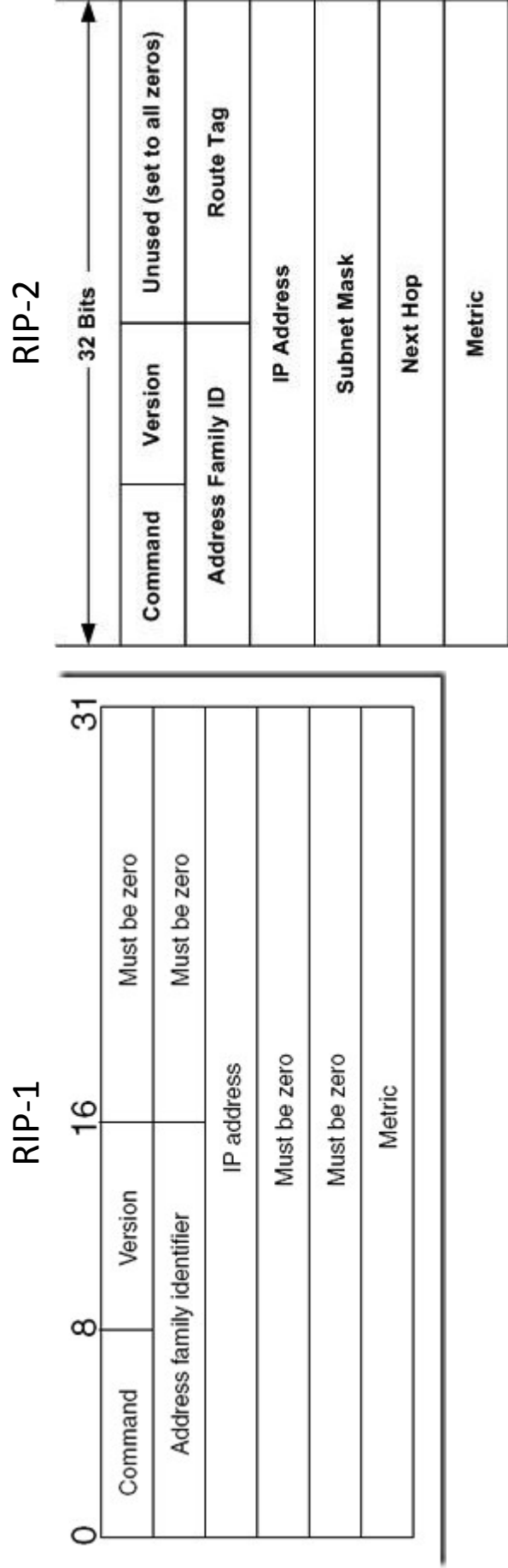
- Technique de l'intoxication d'information (poison reverse)
 - On prévient le voisin duquel on a appris la route d'une route vers la destination avec une distance infinie
 - Méthode agressive qui permet de supprimer les boucles directes
 - Les boucles indirectes persistent par contre...

Le protocole RIP

- RIP (Routing Internet Protocol) est une mise en œuvre de l'algorithme de Bellman-Ford
 - On parle également d'algorithme distribué de Bellman-Ford
 - Utilisé dès l'époque d'ARPANET
- 3 versions du protocole
 - RIP-1 [RFC 1058], RIP-2 [RFC 2453], RIPv2 [RFC 2080]
- La métrique retenue dans RIP est le nombre de sauts
 - Compris entre 1 et 15 et où 16 représente l'infini
 - Ce protocole n'est utilisable qu'à l'intérieur d'un domaine peut étendu
- Les informations sont échangées toutes les 30s entre le routeur et ses voisins
 - Si aucune nouvelle d'un routeur pendant 180s, on considère que la route est rompue

Le protocole RIP

- Le paquet RIP est encapsulé à l'intérieur d'un paquet UDP (port 520)



- Principales nouveautés de RIP-2
 - Support de la notation CIDR
 - Remplacement de la diffusion broadcast des routes de RIP-1 par du multicast (adresse 224.0.0.9)

Le protocole RIP

- Avantages de RIP
 - S'adapte rapidement à une panne sur le réseau
 - Très utilisé
- Principales limitations
 - La distance ne tient compte que du nombre de sauts
 - Pas de prise en compte de la bande passante, du délais, de l'occupation d'un lien...
 - Distance max = 15
 - Trafic de signalisation très important
 - Pas d'authentification pour RIP-1

L'algorithme de Dijkstra

- L'algorithme de Dijkstra (1959) est un algorithme par état de liens qui sert à résoudre le problème du plus court chemin
 - Exemple classique du Voyageur de commerce
 - Il s'applique à un graphe connexe dont le poids de chaque arc est un entier positif
- La topologie du réseau et le coût des liens sont partagés par tous les nœuds
 - Différent de la vision fragmentée de Bellman-Ford
 - Routage par inondation de l'état des liens
- L'algorithme consiste à déterminer pour chaque nœud le plus court chemin de ce nœud vers tous les autres
 - Détermine la table de routage locale en fonction des connaissances partagées

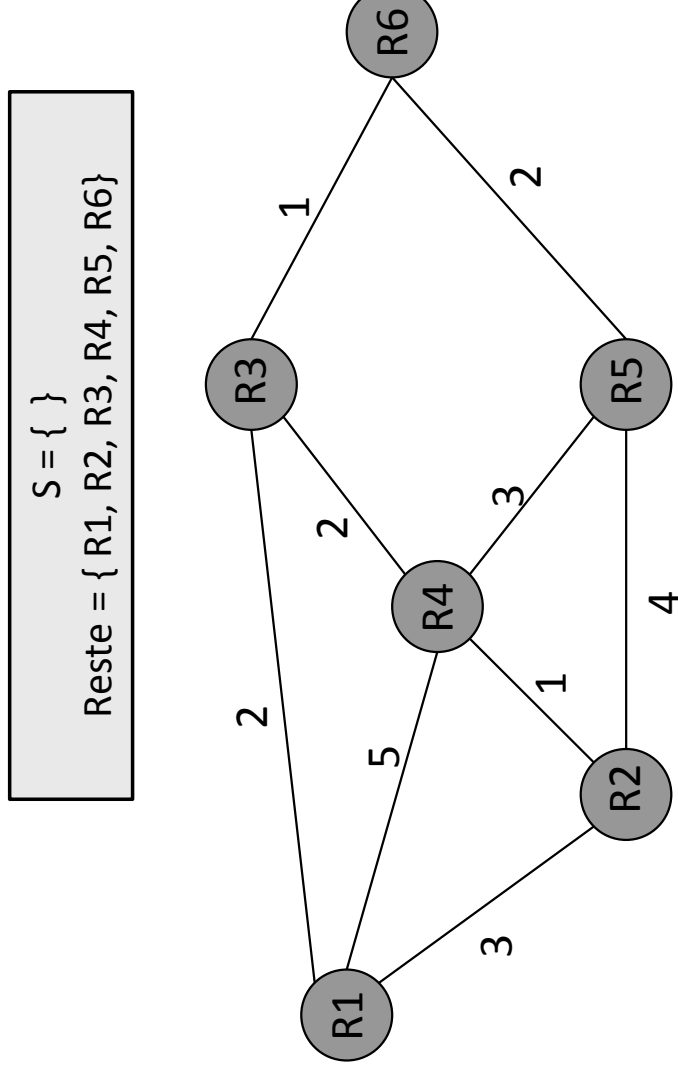
L'algorithme de Dijkstra

➤ Principe de l'algorithme

- Prérequis: La connaissance du réseau et des poids des liens sont connus
- A l'étape 1, on considère le routeur de départ avec un poids de 0
- On évalue ensuite le poids de chacun de ses voisins
- Le voisin le plus proche est ajouté à l'ensemble de recherche (noté S)
- On recherche ensuite le voisin le plus proche des nœuds du graphe et on l'ajoute à l'ensemble de recherche
- On répète l'opération tant que tous les nœuds du réseau n'ont pas été ajoutés

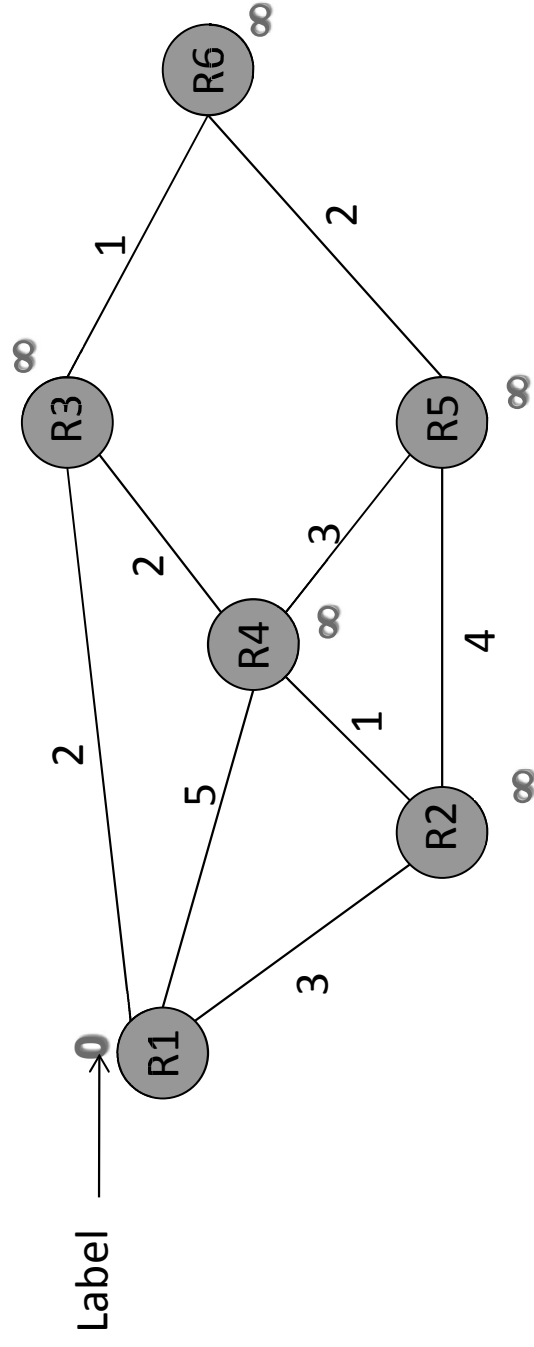
L'algorithme de Dijkstra

- Exemple pour trouver un chemin de R1 à R6



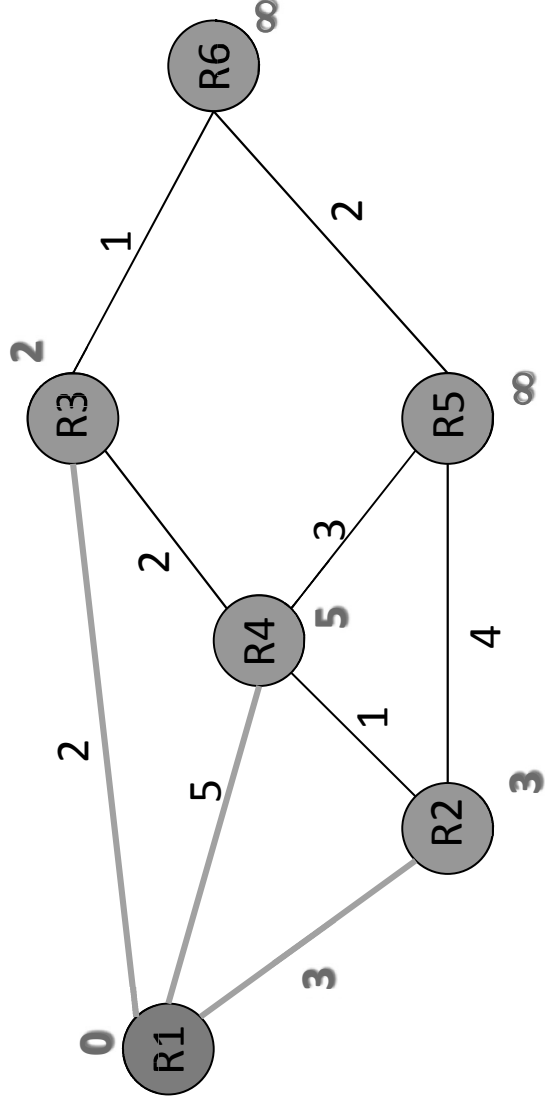
L'algorithme de Dijkstra

$S = \{ \}$
 $\text{Reste} = \{ R1, R2, R3, R4, R5, R6 \}$



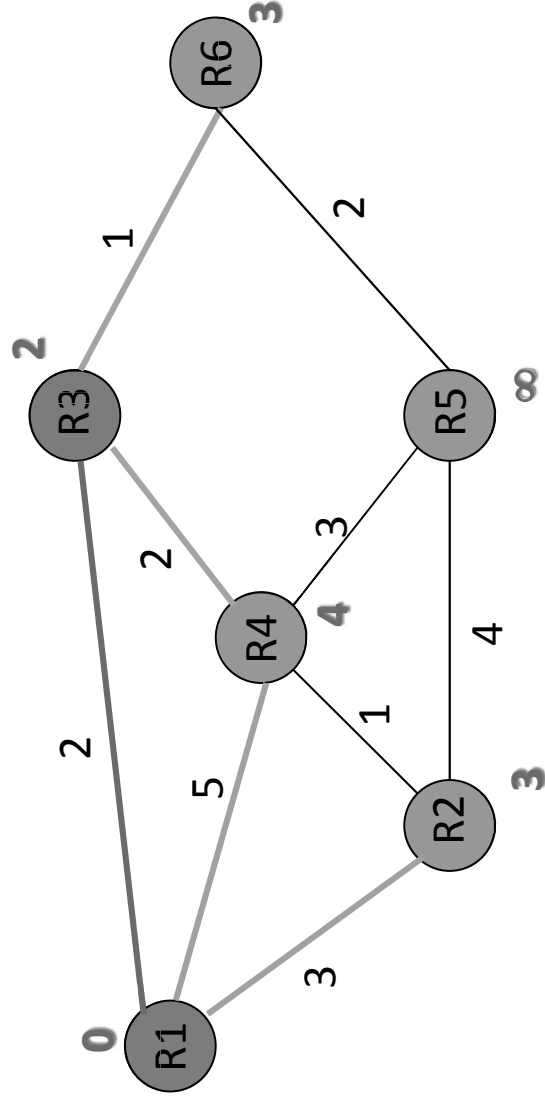
L'algorithme de Dijkstra

$S = \{ R1 \}$
 $Reste = \{ R2, R3, R4, R5, R6 \}$



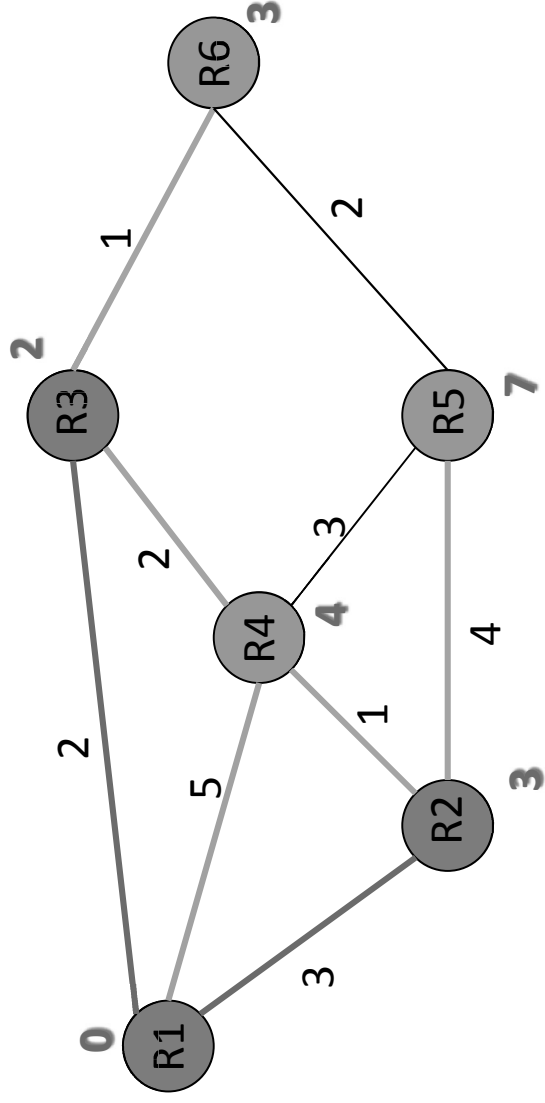
L'algorithme de Dijkstra

$S = \{ R1, R3 \}$
 $Reste = \{ R2, R4, R5, R6 \}$



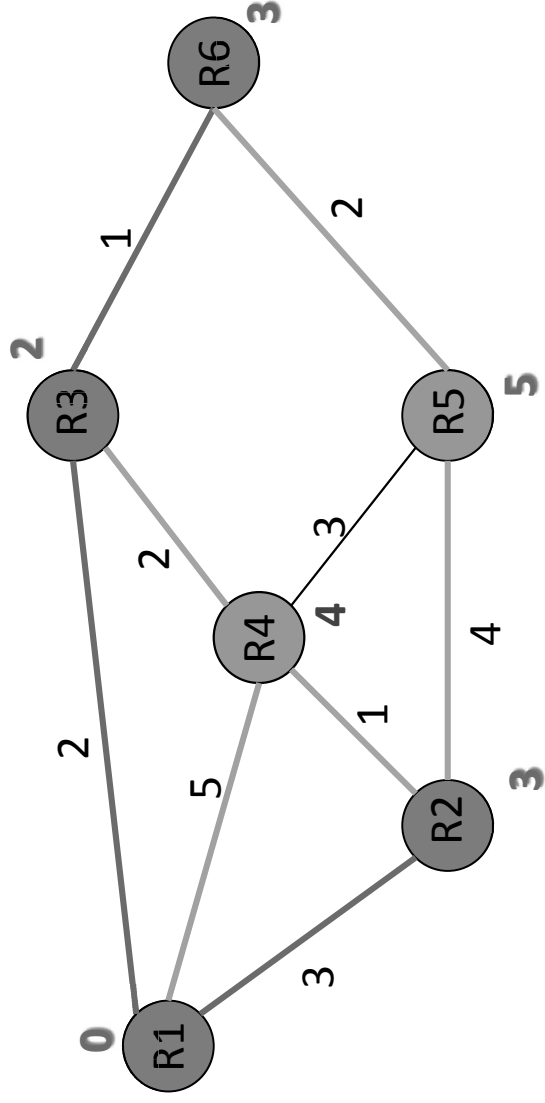
L'algorithme de Dijkstra

$S = \{R1, R3, R2\}$
Reste = $\{R4, R5, R6\}$



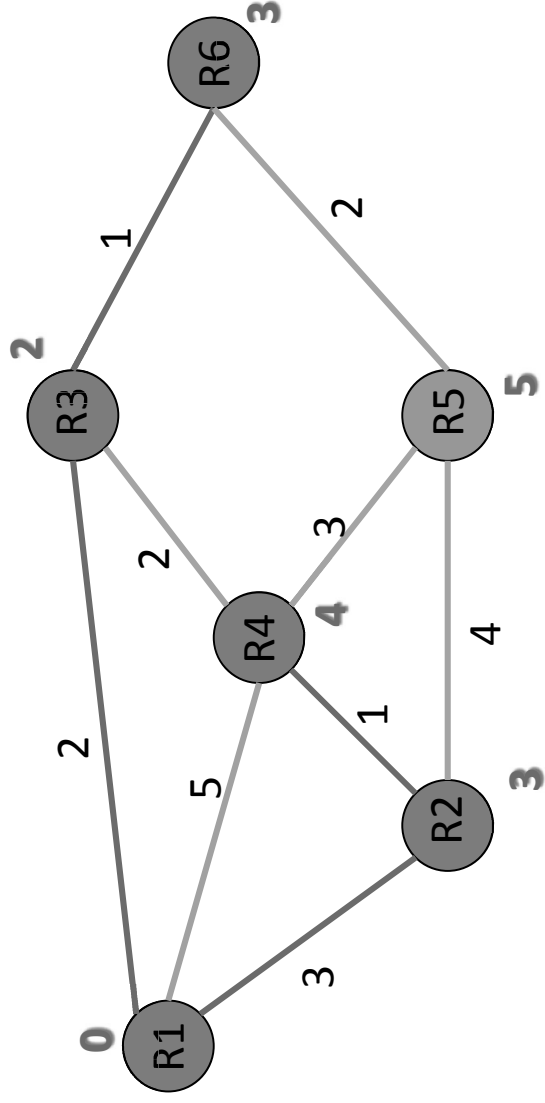
L'algorithme de Dijkstra

$S = \{ R1, R3, R2, R6 \}$
Reste = $\{ R4, R5 \}$



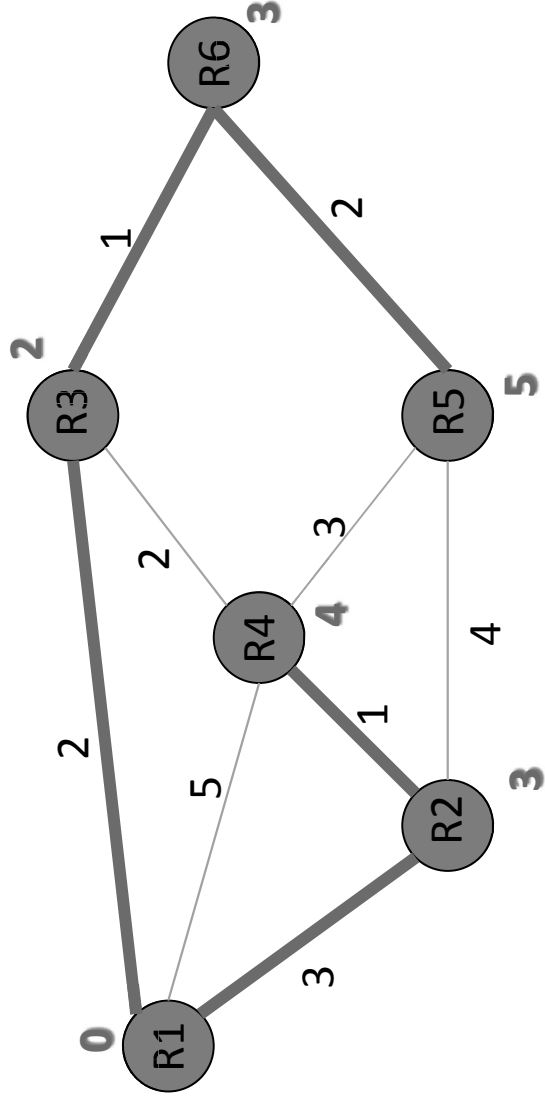
L'algorithme de Dijkstra

$S = \{ R1, R3, R2, R6, R4 \}$
Reste = $\{ R5 \}$



L'algorithme de Dijkstra

$S = \{ R1, R3, R2, R6, R4, R5 \}$
Reste = $\{ \}$



Le protocole OSPF

- Ce protocole est une des applications les plus courantes de l'algorithme de Dijkstra
 - Défini dans [RFC 2328] et dans [RFC 5340] pour la version 3 qui supporte IPv6
- Il a été conçu pour répondre à la plupart des limitations de RIP
 - Possibilité de gérer plus de 16 routeurs
 - Plus rapide à converger vers une solution
 - Métrique plus fine que RIP (prise en compte des débits)
 - Segmentation possible du domaine en aires
 - ...
- Par contre OSPF est plus complexe que RIP à mettre en œuvre
 - Nécessite des routeurs plus puissants

Le protocole OSPF

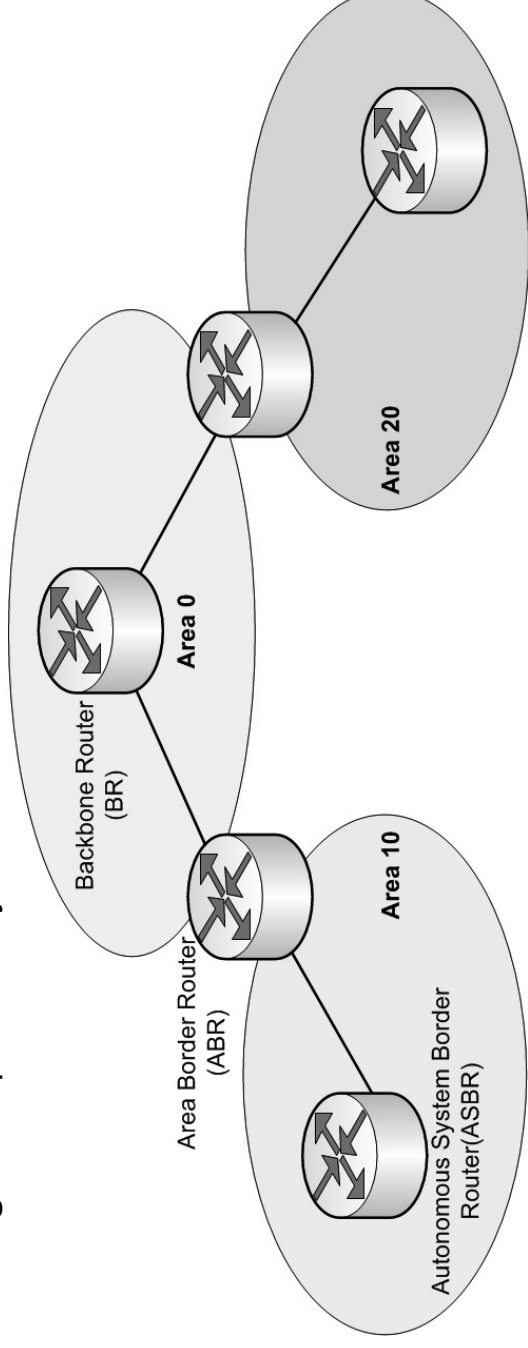
- OSPF est donc adapté à des réseaux de grande taille
 - Il oblige à structurer son réseau
 - Attribut un poids aux liens (aussi appelé coût)
 - Le poids est défini suivant la formule : $poids = \frac{10^8}{\text{bande passante en b/s}}$
 - Ex: pour un réseau 100Mb/s le poids est de 1
 - Principe de fonctionnement
 - Chaque routeur connaît ou peut identifier ses voisins
 - S'il y a plusieurs routeurs sur un réseau, un routeur principal est élu
 - Le routeur principal va fournir sa base de donnée aux autres routeurs
 - Chaque routeur diffuse à ses voisins (messages Link State Advertisment ou LSA)
 - La liste de ses voisins immédiats
 - Le poids des liens vers ses voisins
-

Le protocole OSPF

- Chaque routeur met à jours sa base de données et doit obtenir à la fin du processus une vision complète du réseau
 - Chaque routeur calcule de manière autonome les meilleures routes à emprunter grâce à l'algorithme de Dijkstra et construit les tables de routage correspondantes
 - On choisit toujours en premier la branche avec le poids minimum
- Lorsqu'un changement se produit dans le réseau, chaque routeur se met à diffuser sa table d'états de lien
- La recherche de routes est ensuite relancée pour tous les routeurs
- Dans le cas d'un très grand domaine OSPF, les changements peuvent être fréquents
- Consommation importante de bande passante et de CPU au niveau des routeurs
 - La modification peut être mineure
-

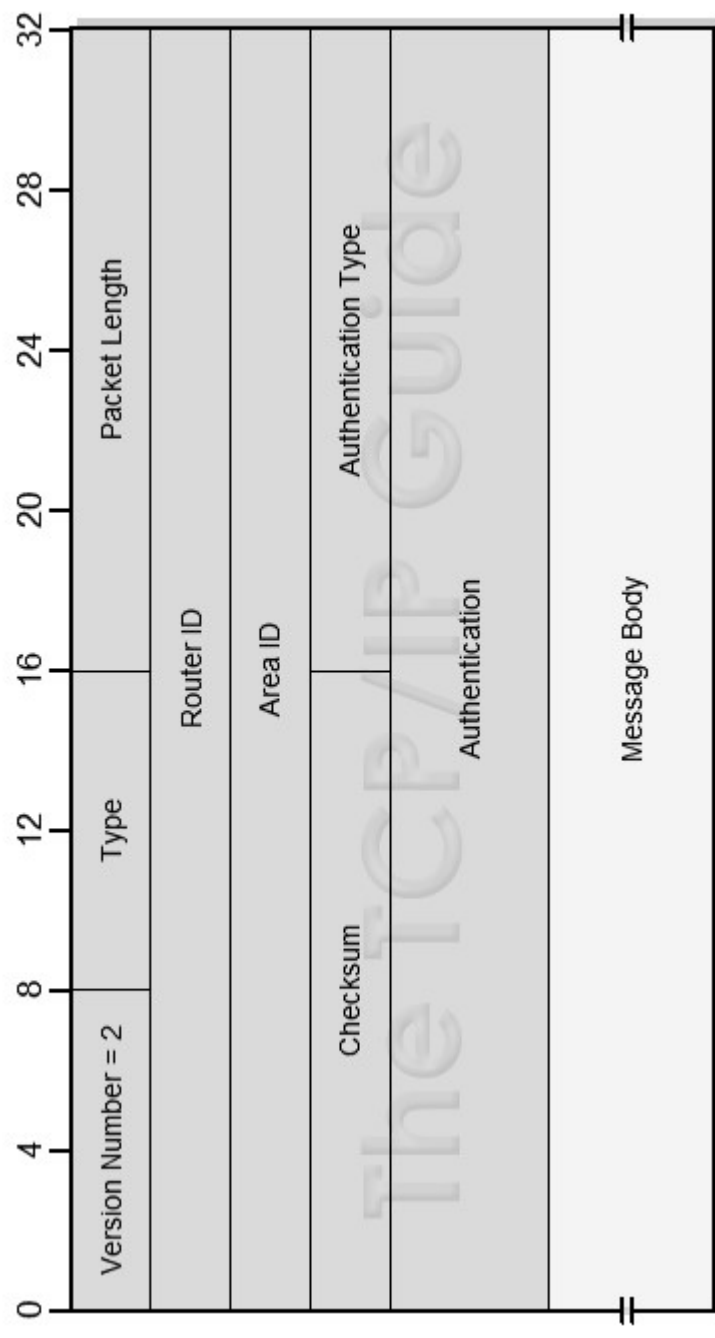
Le protocole OSPF

- Pour cloisonner ses modifications, OSPF introduit la notion d'Aires
 - Une aire est plus simple et donc plus stable
 - Les aires sont numérotées
 - Les modifications sont contenues au niveau d'une aire
 - Une aire globale (backbone) relie toutes les aires entre elles
 - Elle connait toutes les informations de routage mais ne diffuse que des condensés
- ★ *Un condensé va contenir des routes agrégées par exemple*
 - L'aire globale prend toujours le numéro 0



Le protocole OSPF

➤ En-tête OSPF

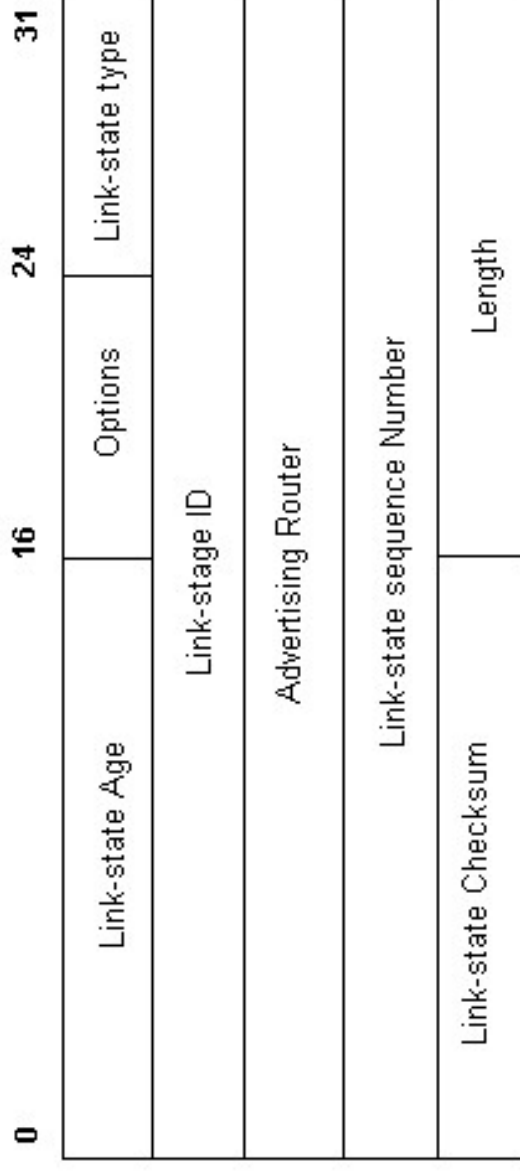


Le protocole OSPF

- OSPF définit 5 types de paquets différents
 - HELLO (type 1)
 - Maintenance des liens et échange d'informations entre routeurs
 - Database description (type 2)
 - description des LSA
 - Link State Request (type 3)
 - Pour obtenir un LSA
 - Link State Update (type 4)
 - Pour la mise à jours d'un LSA
 - Link State Acknowledgment (type 5)
 - Pour acquitter la réception d'un Link State Update

Le protocole OSPF

- Les link state advertisement (LSA) possèdent également un format particulier



- On compte 7 types différents pour
- Décrire l'état des interfaces d'un routeur
 - Décrire les routeurs attachés à un sous-réseau
 - Envoyer un condensé de route pour un ABR
 - ...

Autonomous System

- L'ensemble des routeurs administrés par un même ISP va définir **un système autonome (AS)** dans Internet
- Les protocoles que nous avons vus ne sont utilisables qu'à l'intérieur d'un AS
 - On parle de routage interne
- Pour effectuer du routage entre AS on va parler de routage externe et des protocoles dédiés comme Border Gateway Protocol (BGP) existent
- BGP est le standard pour internet
 - Repose sur la notion de vecteur de chemin
 - Il utilise TCP pour l'échange de sa signalisation