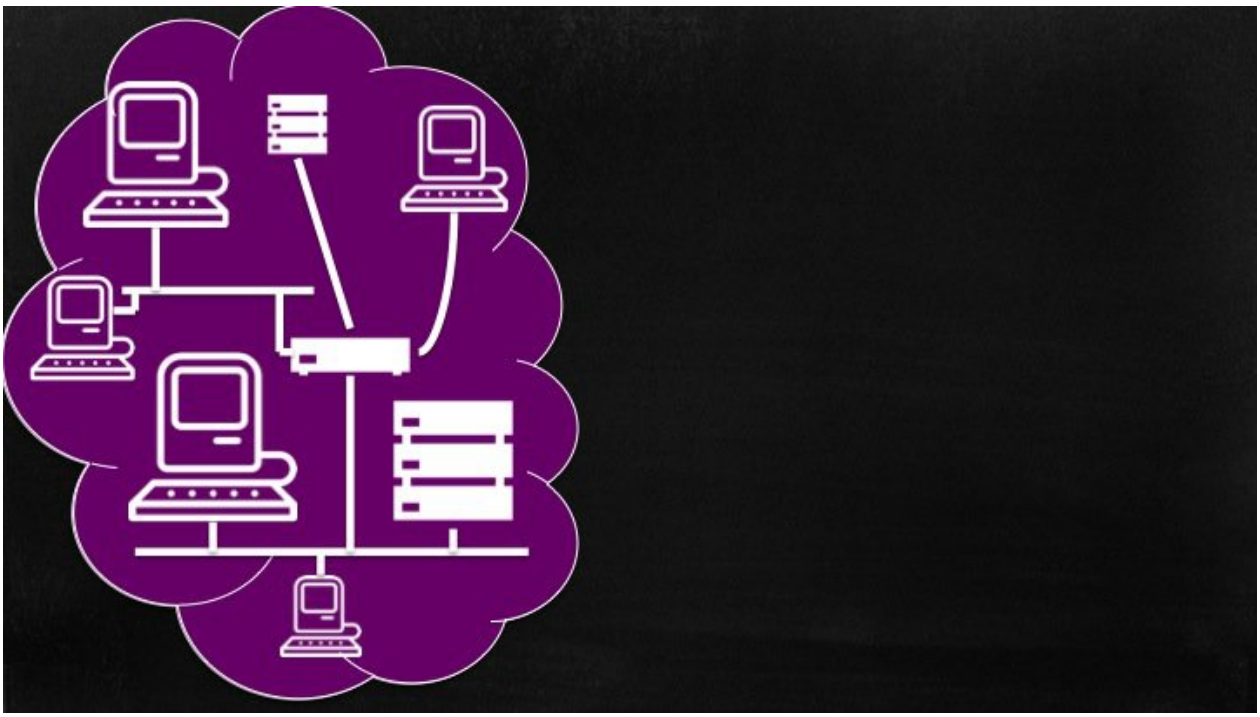
Semaine 5 : le routage

Séance 1 : la fonction de relayage

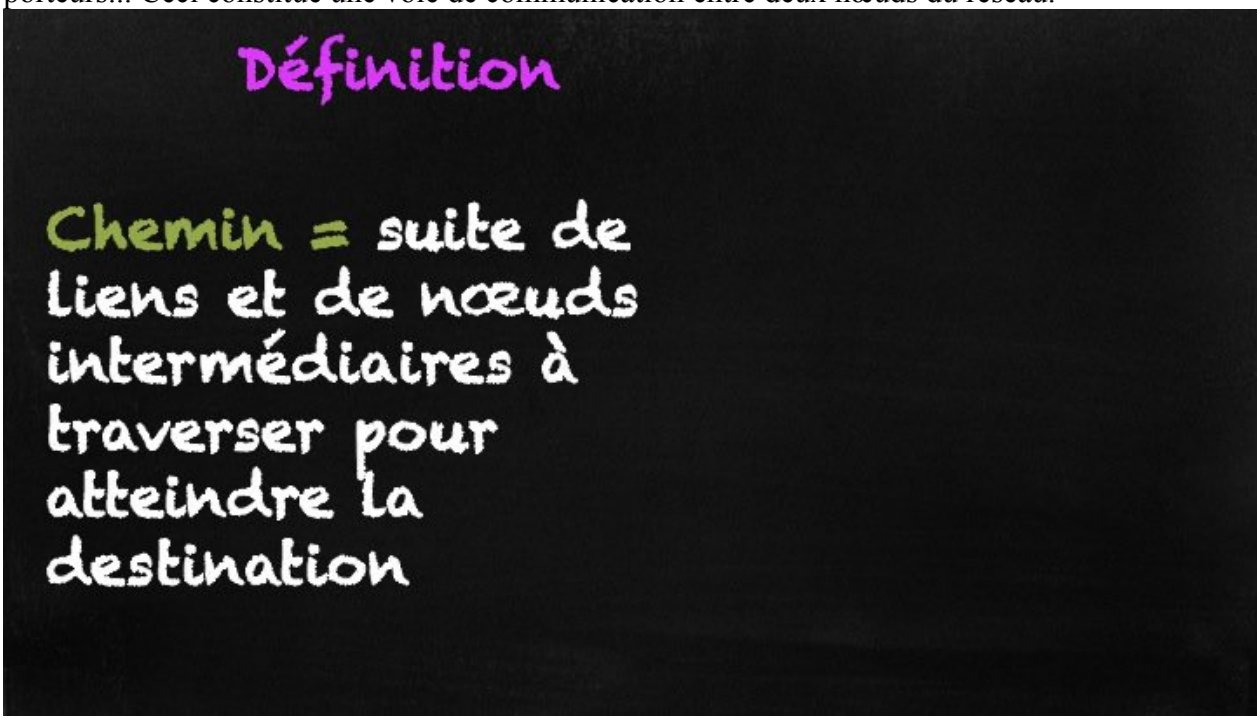
Introduction

Aujourd'hui, nous allons voir comment l'adresse est utilisée pour acheminer le trafic dans le réseau.

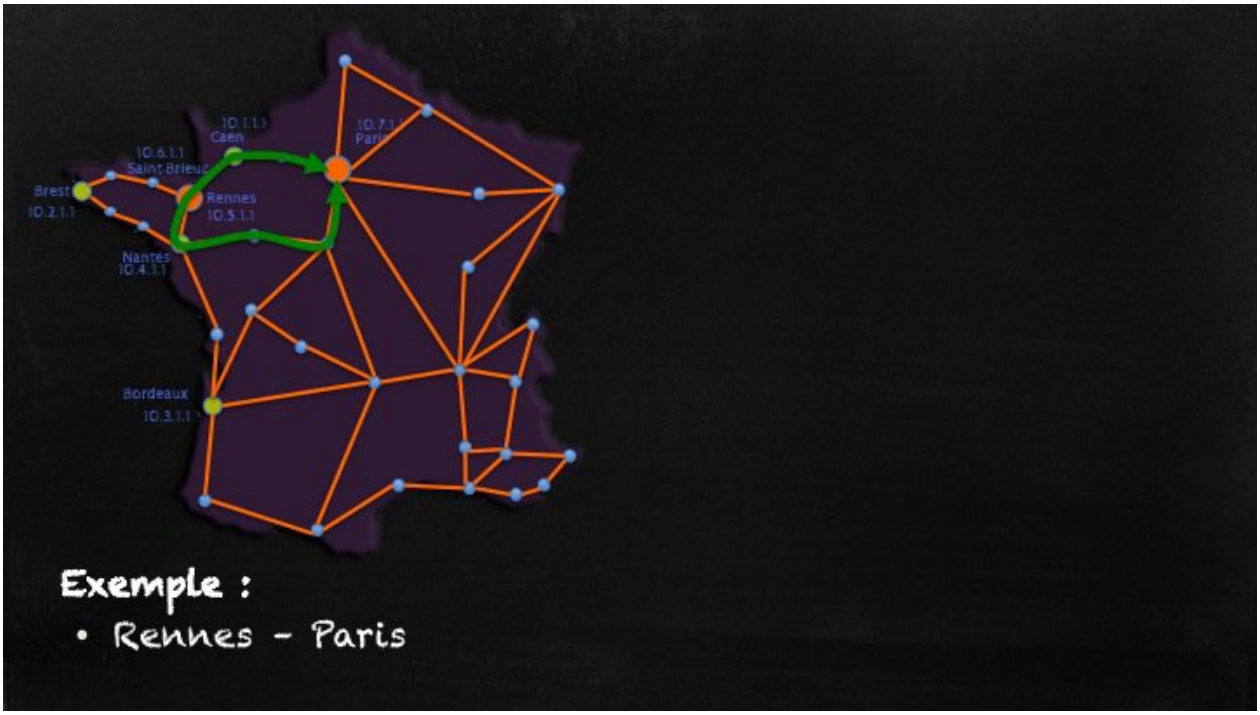
Notion de chemin



Un réseau est constitué d'un ensemble d'équipements et de machines reliées pour pouvoir communiquer, par exemple, par des câbles, des fibres optiques, des liaisons sans fil, les courants porteurs... Ceci constitue une voie de communication entre deux nœuds du réseau.



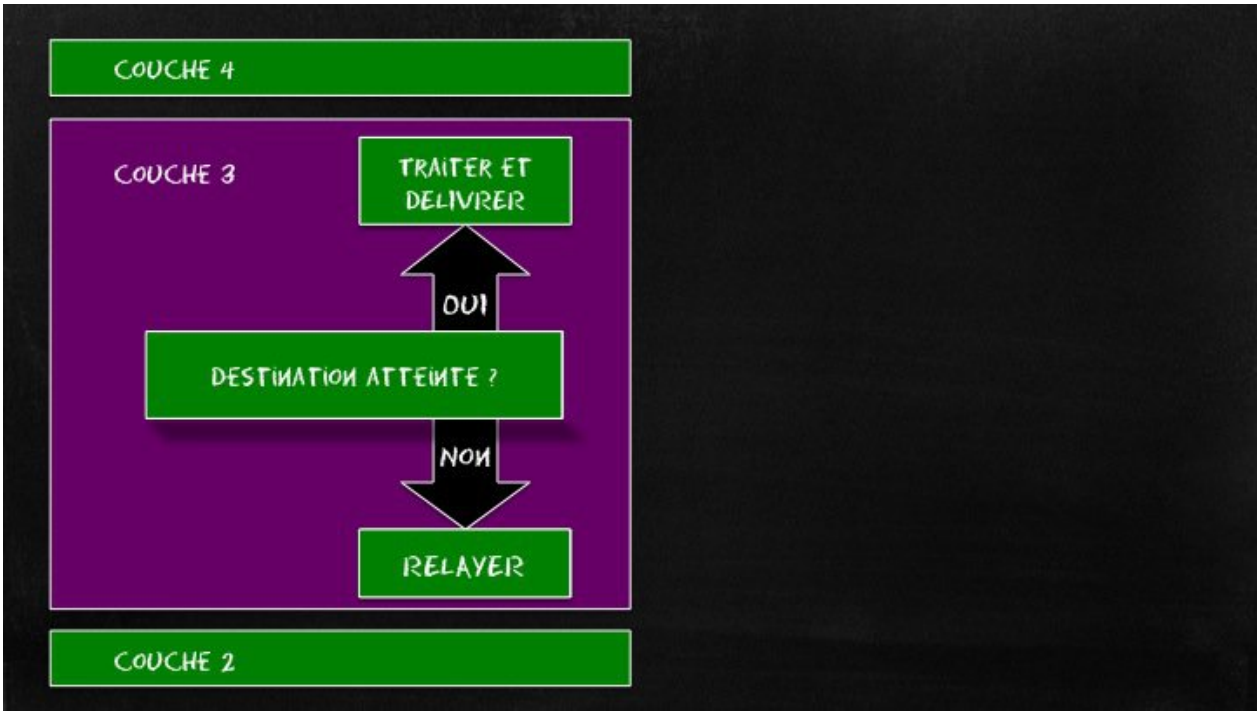
On peut définir la notion de chemin dans le réseau par la suite de liens et de nœuds intermédiaires parcourus pour aller de la source à la destination dans le réseau.



Prenons l'exemple de ce réseau fictif. Vous voyez une carte de France avec des nœuds qui représentent des endroits où on trouve un équipement réseau important et des liaisons qui joignent ces différents équipements. Si l'on souhaite envoyer du trafic de Rennes à Paris, il existe plusieurs chemins pour joindre Paris à partir de Rennes, qui passent par exemple par :

- Rennes Caen Rouen et Paris
- ou Rennes Nantes Angers Orléans Paris

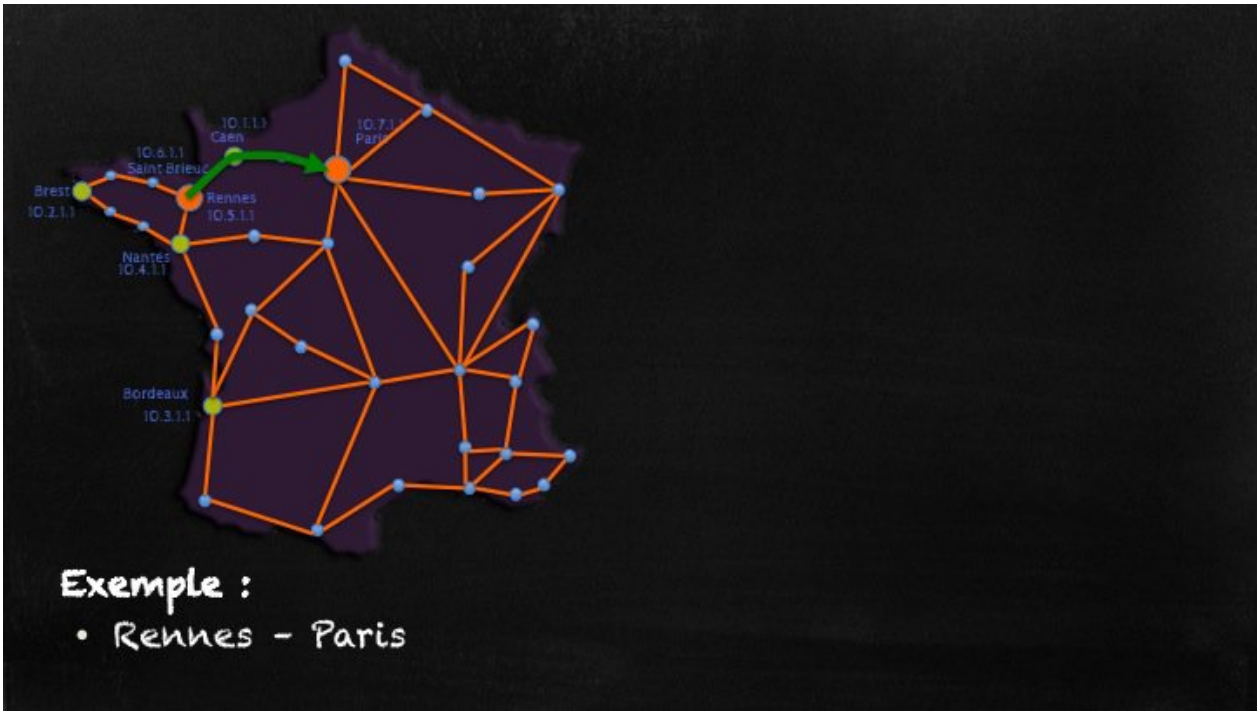
Dans les réseaux, nous allons en général garder un chemin principal pour acheminer les données et nous allons voir comment nous allons parcourir ce chemin grâce à la fonction de relayage. Il existe de nombreuses méthodes pour acheminer les données. Nous ne traiterons ici que le mécanisme utilisé par IP. Vous trouverez dans le livre, au chapitre 6, d'autres méthodes : circuit virtuel, routage par la source…



Dans IP, lors du relayage, on dit que l'acheminement est fait par la destination. Cela veut dire que l'on va considérer uniquement l'adresse de la destination pour savoir vers où envoyer la donnée pour qu'elle atteigne la destination finale.

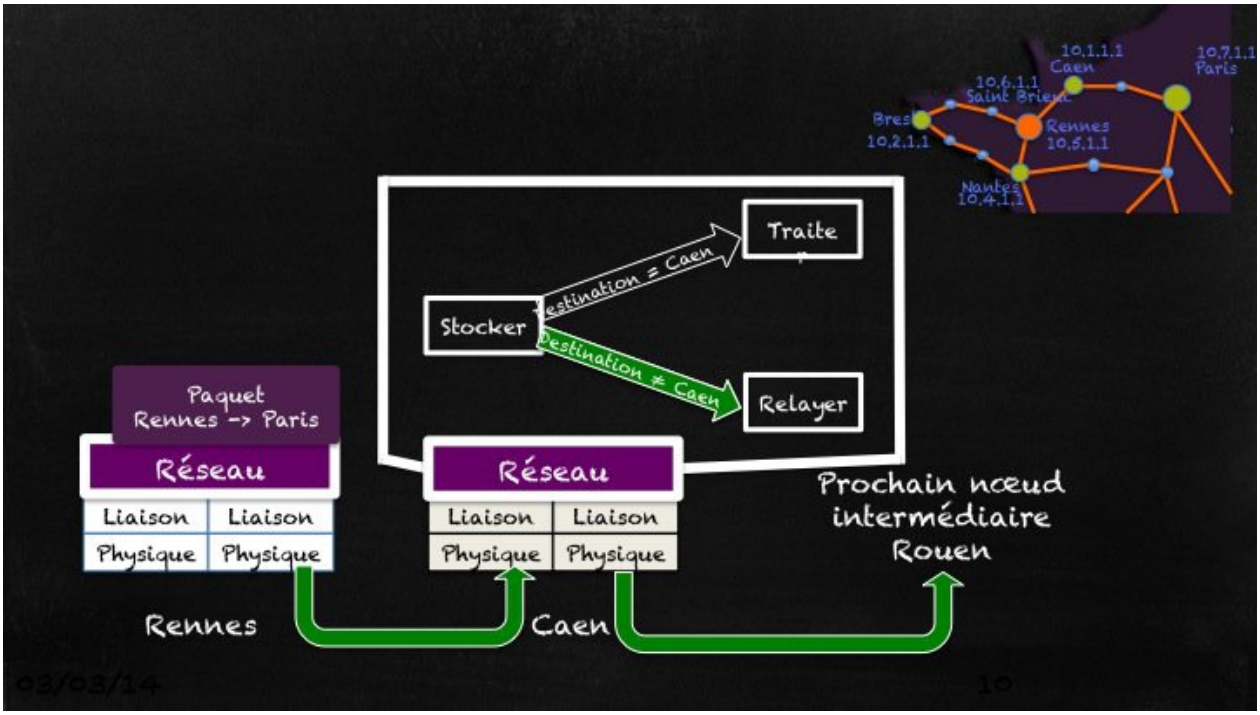
Chaque nœud intermédiaire va exécuter un algorithme assez simple :

- regarder d'abord le paquet reçu et considérer son adresse destination pour la comparer à l'adresse du nœud
    - Si elles correspondent, le paquet est arrivé à la destination, qui va pouvoir le traiter.
    - Si elles ne correspondent pas, il va falloir choisir quel est le prochain saut et envoyer le paquet vers ce prochain équipement.



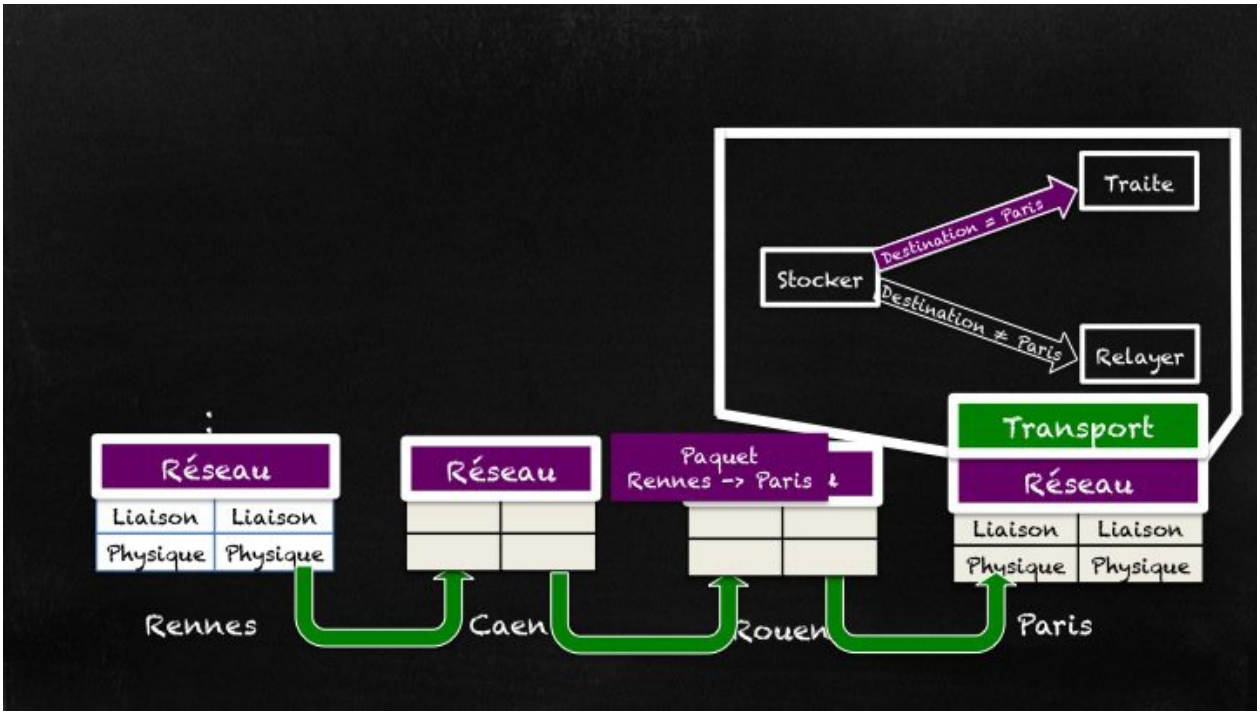
Reprenons l'exemple de la communication entre Rennes et Paris sur le réseau fictif que nous avons vu précédemment.

Le chemin sélectionné par le réseau est Rennes-Rouen-Caen-Paris.



Rennes étant la source, le paquet (PDU IP) descend dans les couches protocolaires pour être émis sur le support physique. Le paquet arrive à Caen. Le paquet remonte dans les couches

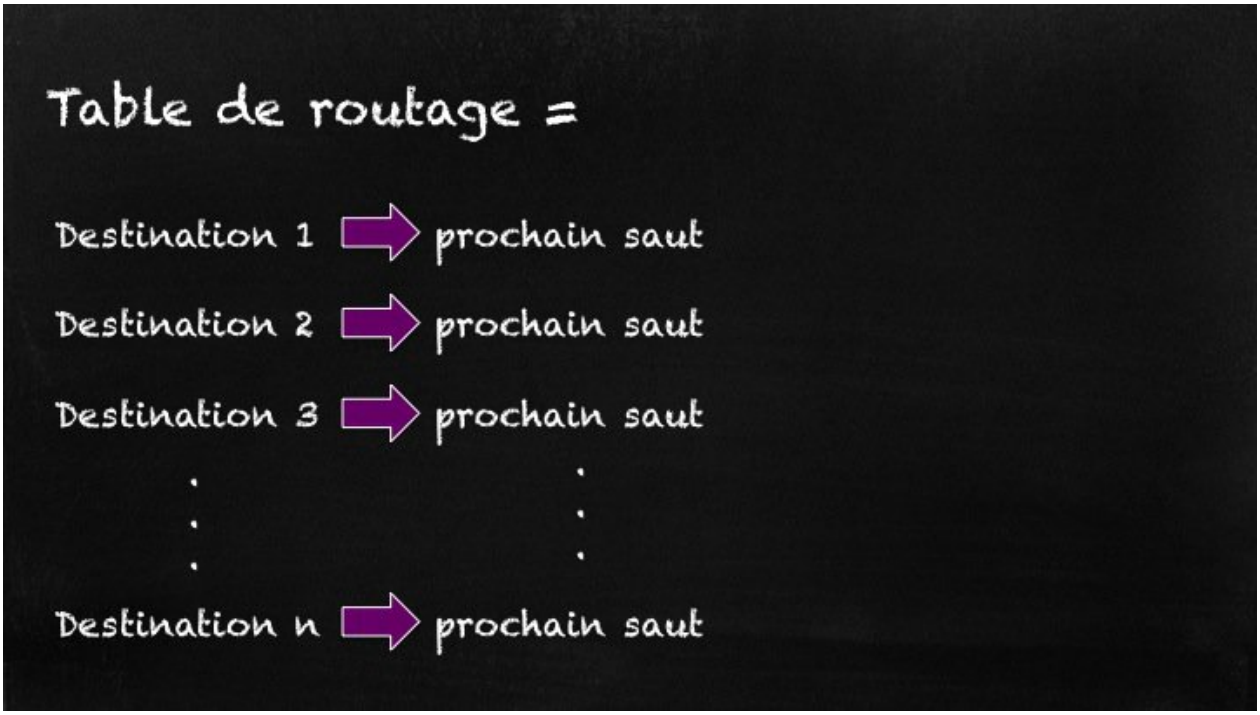
protocolaires jusqu'à la couche réseau qui examine l'adresse destination et exécute l'automate suivant : l'entité protocolaire IP stocke le paquet et compare son adresse propre à celle de la destination indiquée dans le paquet. Comme il n'y a pas correspondance, l'entité IP relaye le paquet vers le prochain saut : Rouen. Le paquet redescend alors à travers les couches 2 et 1 pour être transmis sur le lien entre Caen et Rouen.



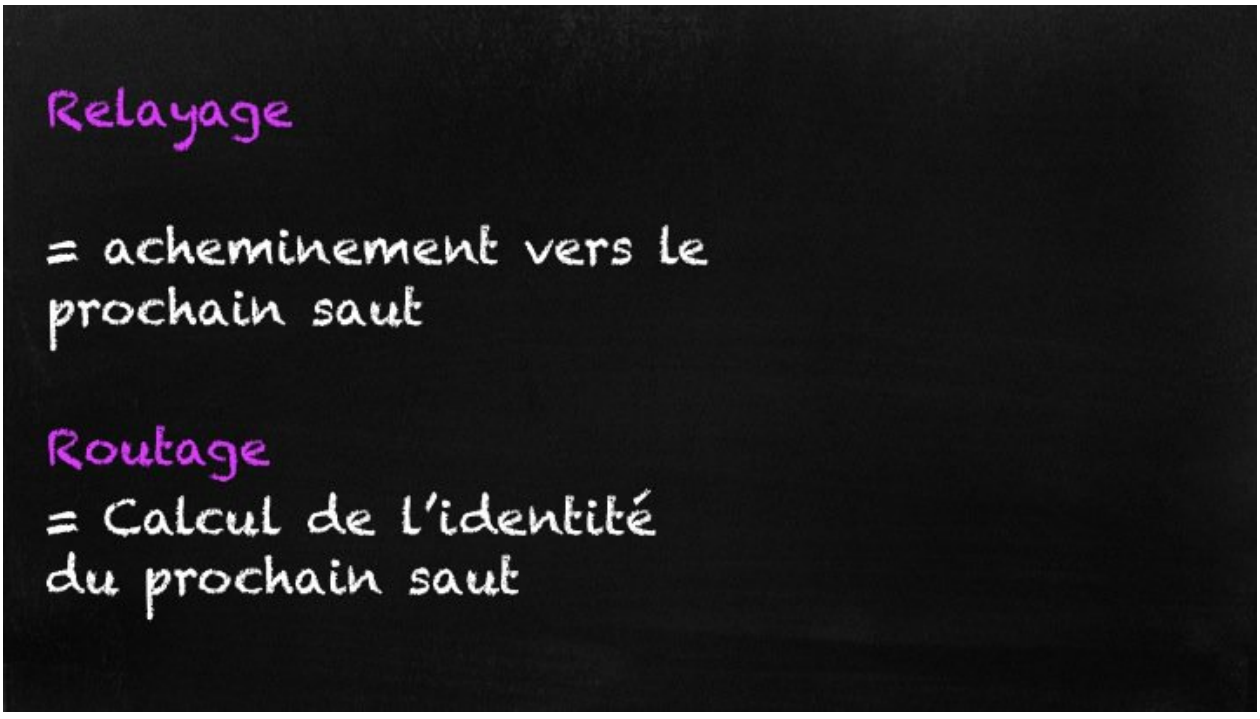
Le paquet arrive maintenant à Rouen. Le même algorithme sera exécuté par l'équipement de Rouen. On dit qu'il transite par Rouen.
L'entité IP (par simplification, on dit aussi l'équipement) constate que la destination n'est pas Rouen et le transmet vers le prochain saut : Paris. Arrivé à Paris, le paquet remonte jusqu'à la l'entité IP de la couche réseau qui s'aperçoit que le paquet a atteint sa destination. Il sera donc délivré à la couche de niveau supérieur sur le SAP indiqué dans le champ Protocole du paquet IP (en général une entité de la couche transport).

La table de routage

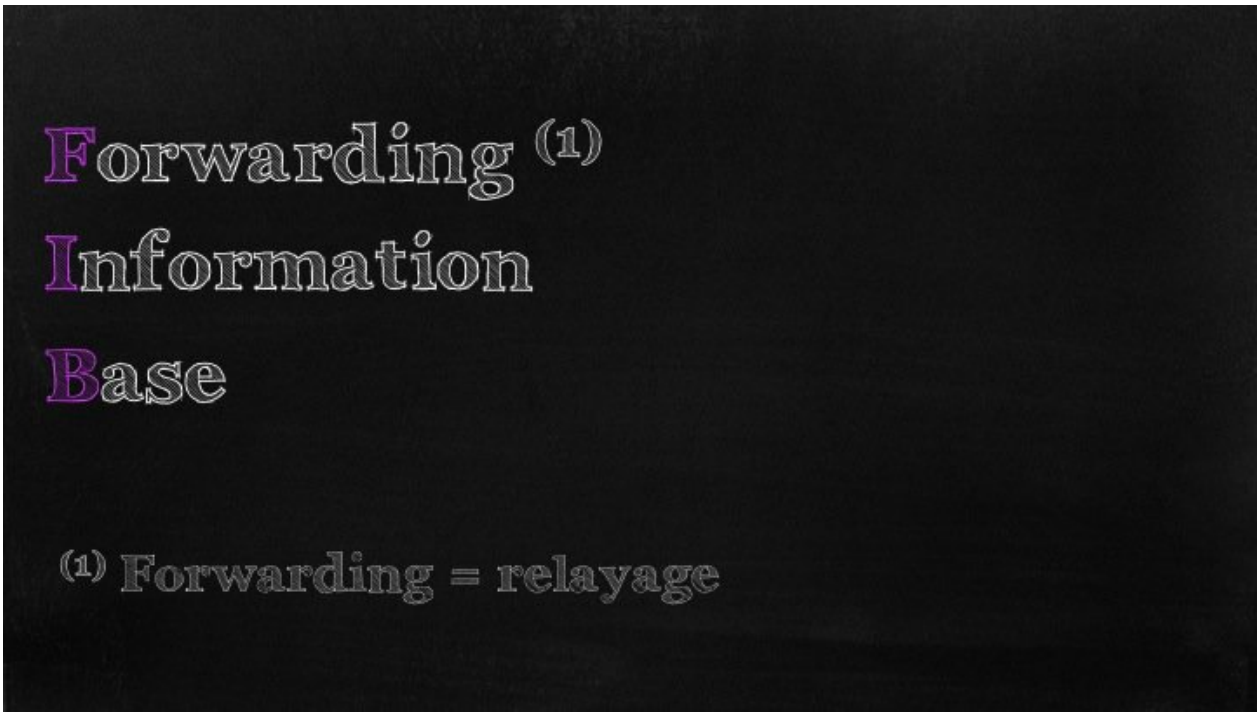
Dans l'exemple précédent, nous avons considéré que le nœud intermédiaire connaît le prochain saut à utiliser pour joindre la destination. Où trouve-t-il cette information ? Dans la table de routage.



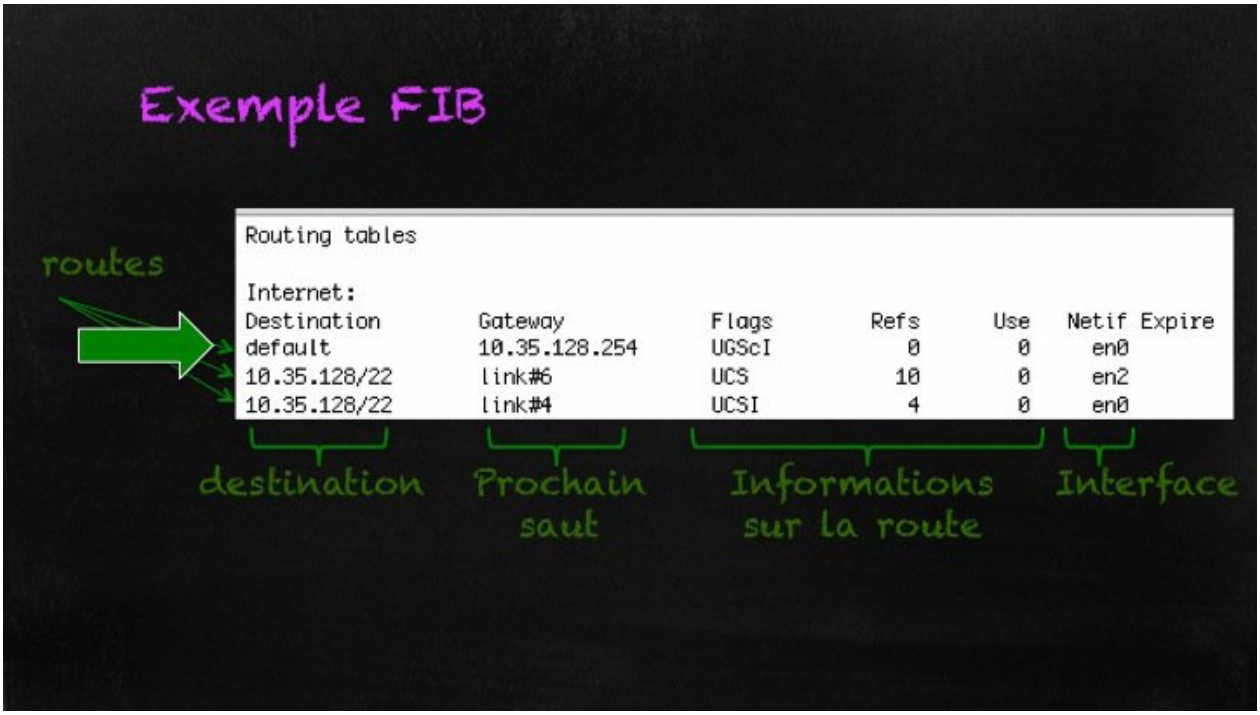
Chaque nœud intermédiaire dispose d'une table locale qui associe, pour chaque destination connue, le prochain saut à utiliser ; un peu comme dans un carrefour routier où les panneaux indicateurs donnent les directions pour vous diriger vers votre destination. Les panneaux indicateurs constituent la table de routage en un nœud du réseau routier. La destination peut être désignée dans la table de routage IP soit par son adresse IP si c'est une machine, soit par le préfixe du réseau auquel elle appartient.



Le terme de table de routage est ambigu. Il cache deux notions : Le relayage : la table est utilisée par la fonction de relayage de l'entité IP pour acheminer le paquet vers le prochain saut.Le routage : un protocole va calculer les meilleurs chemins (donc, pour chaque nœud le prochain saut) et mettre à jour les informations dans la table (nous verrons cela ultérieurement).

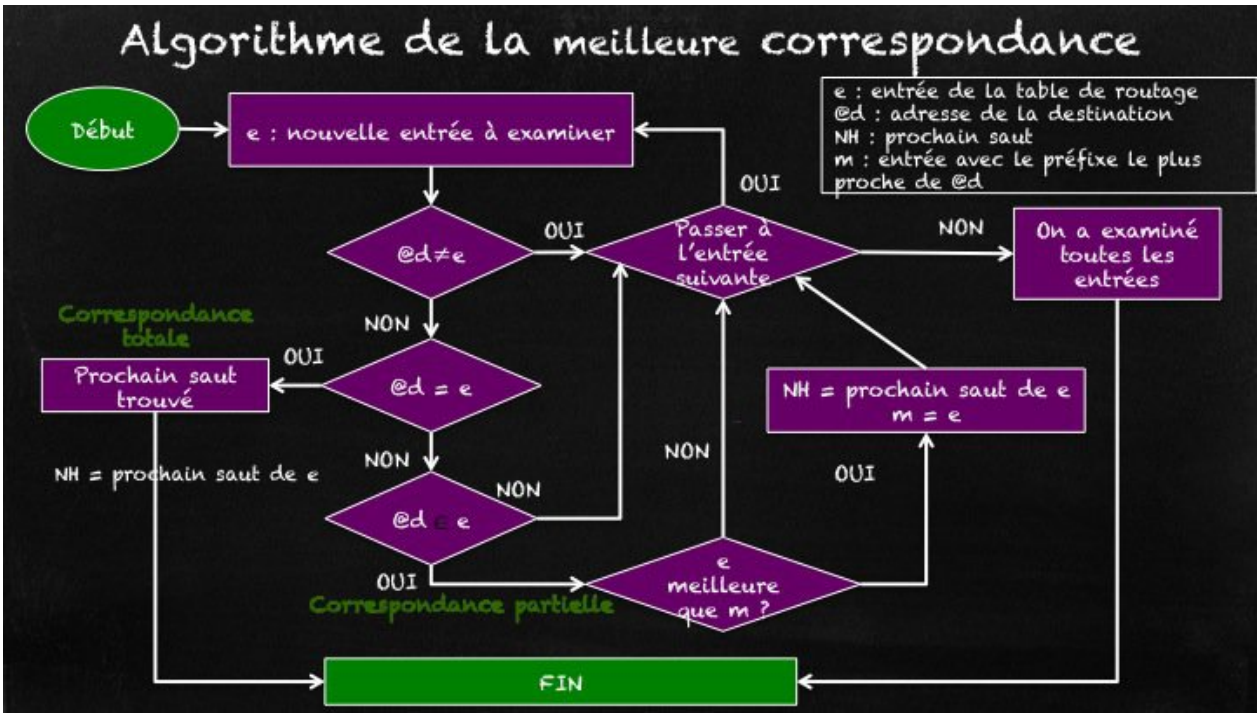


En anglais, on utilise le terme de *FIB* : Forwarding Information Base ; Forwarding voulant dire relayage.



Prenons un exemple de table de routage. Vous voyez ce tableau qui possède plusieurs colonnes et plusieurs lignes. Chaque ligne désigne une destination connue. On va pouvoir trouver la route vers ces équipements connus dans les lignes de la table de routage. La première colonne désigne la destination. On y trouve l'adresse IP d'une machine ou un préfixe réseau. La colonne suivante désigne le prochain saut à utiliser. La dernière colonne indique l'interface (la voie locale) de sortie pour aller vers ce voisin. Vient ensuite une notion de coût associé à la route et divers renseignements dans les colonnes suivantes.
Vous noterez une ligne particulière où on trouve une route par défaut. Si la destination du paquet ne figure pas dans la table de routage, il faut quand même pouvoir l'atteindre. On va donc transmettre ce paquet à un routeur qui a plus d'informations ou de connaissance et qui, on l'espère, saura, lui, l'acheminer correctement. On appelle ce routeur, le « routeur par défaut ».
Dans la table de routage, on a soit des lignes décrivant comment atteindre une destination, soit une entrée pour le routeur par défaut (en quelque sorte le panneau « autres directions » du carrefour routier).

L'algorithme de la meilleure correspondance

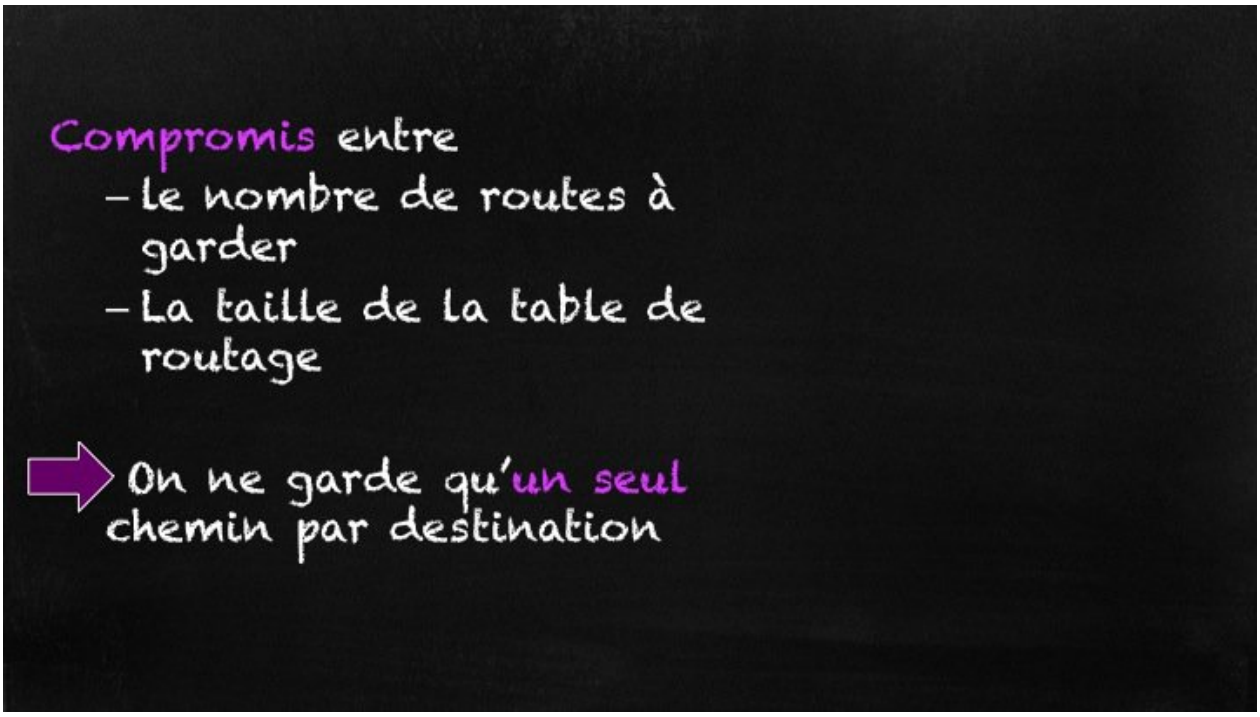


La fonction de relayage utilise la table de routage en appliquant l'algorithme de la meilleure correspondance. Quand on veut transmettre un paquet, on regarde l'adresse de sa destination et on la cherche dans la table de routage. Le routeur va ainsi parcourir toute la table de routage jusqu'à trouver l'adresse de la machine, si elle est connue, ou trouver le préfixe qui correspond le mieux au réseau de la machine.
Imaginons qu'il ne connaisse pas ce réseau. On va parcourir toute la table de routage et s'apercevoir que l'on n'a pas trouvé de correspondance. On utilisera alors l'adresse du routeur par défaut qui, on l'espère, aura une meilleure connaissance d'Internet et pourra transmettre le paquet vers la destination.
On voit bien l'influence de la taille de la table de routage quand on va vouloir transmettre un paquet : plus elle sera grande, plus on passera du temps à la parcourir pour trouver le prochain saut.

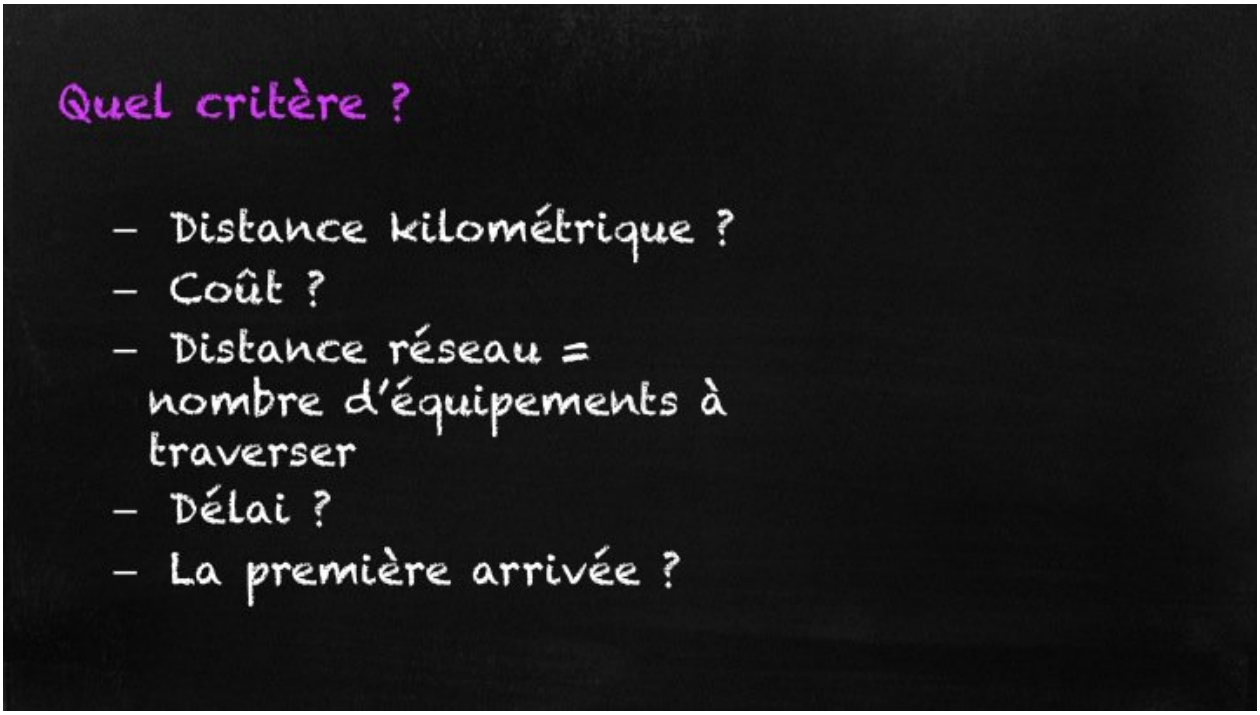
Séance 2 : protocole de routage

Introduction

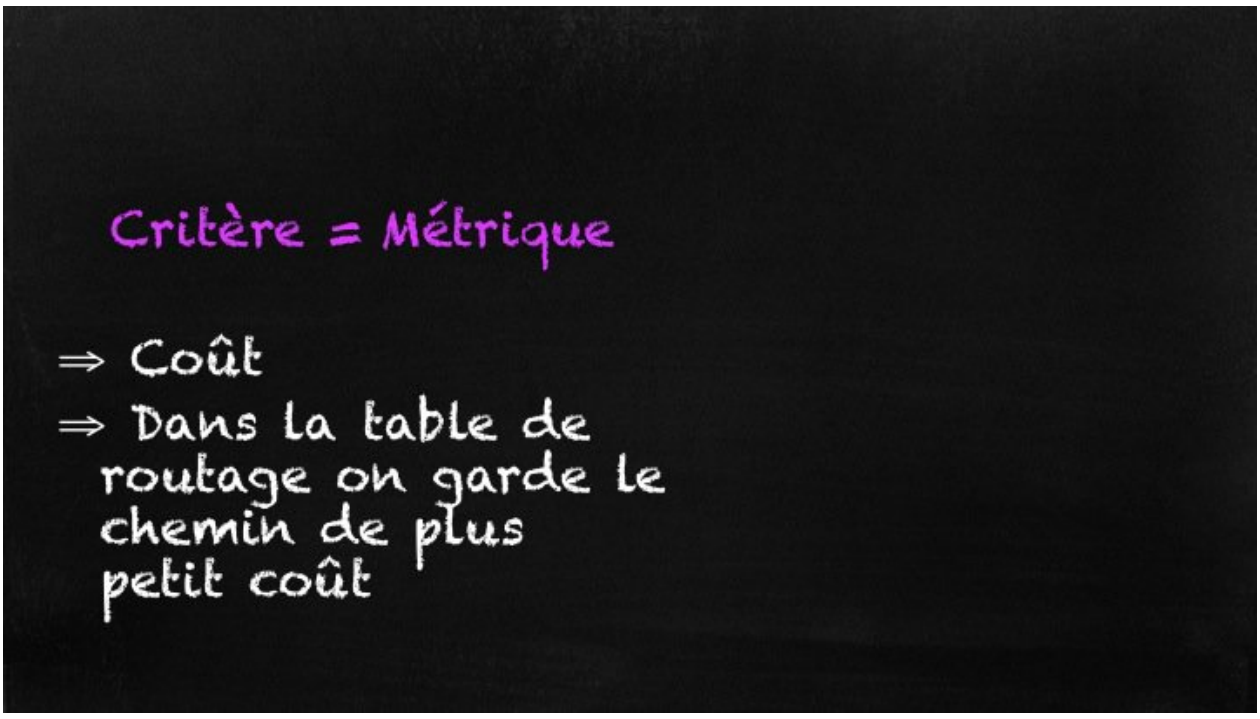
Dans la séance précédente, nous avons vu comment utiliser les informations de la table de routage pour acheminer les paquets dans le réseau. Nous allons maintenant étudier comment calculer, construire, et maintenir les informations de la table de routage grâce au protocole de routage.



Nous avons déjà vu que plusieurs chemins sont disponibles dans le réseau. Nous avons aussi vu que la taille de la table de routage peut être importante et que l'algorithme de la meilleure correspondance a un impact non négligeable sur le temps de traitement des paquets. Le protocole de routage va donc devoir faire un compromis entre le nombre de routes gardées pour chaque destination et la taille de la table de routage. En général, on ne garde qu'un chemin vers chaque destination. Nous verrons plus loin comment organiser le réseau pour limiter la complexité des tables de routage en le décomposant en aires.

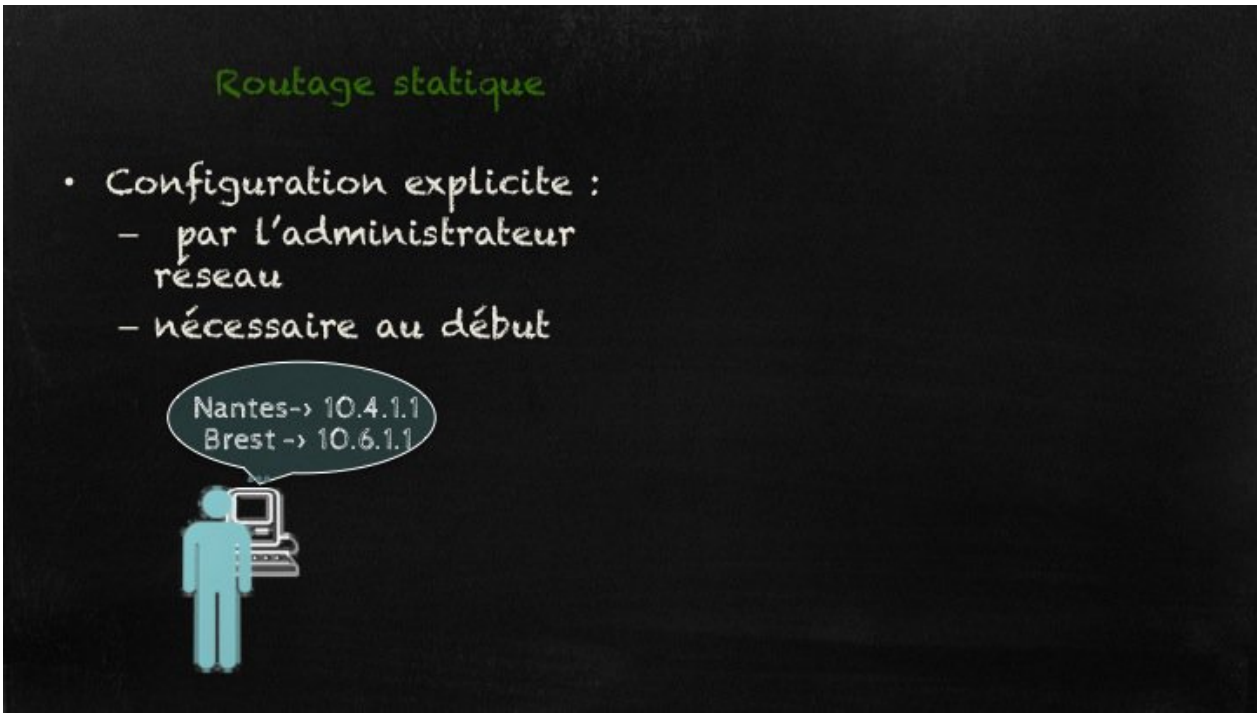


Le protocole de routage va devoir choisir un chemin dans le réseau selon un certain critère. Cela peut être la distance kilométrique, le coût monétaire de chacune des liaisons, la distance réseau (le nombre d'équipements intermédiaires à parcourir sur le chemin), le délai associé au chemin… On peut imaginer toutes sortes de critères.

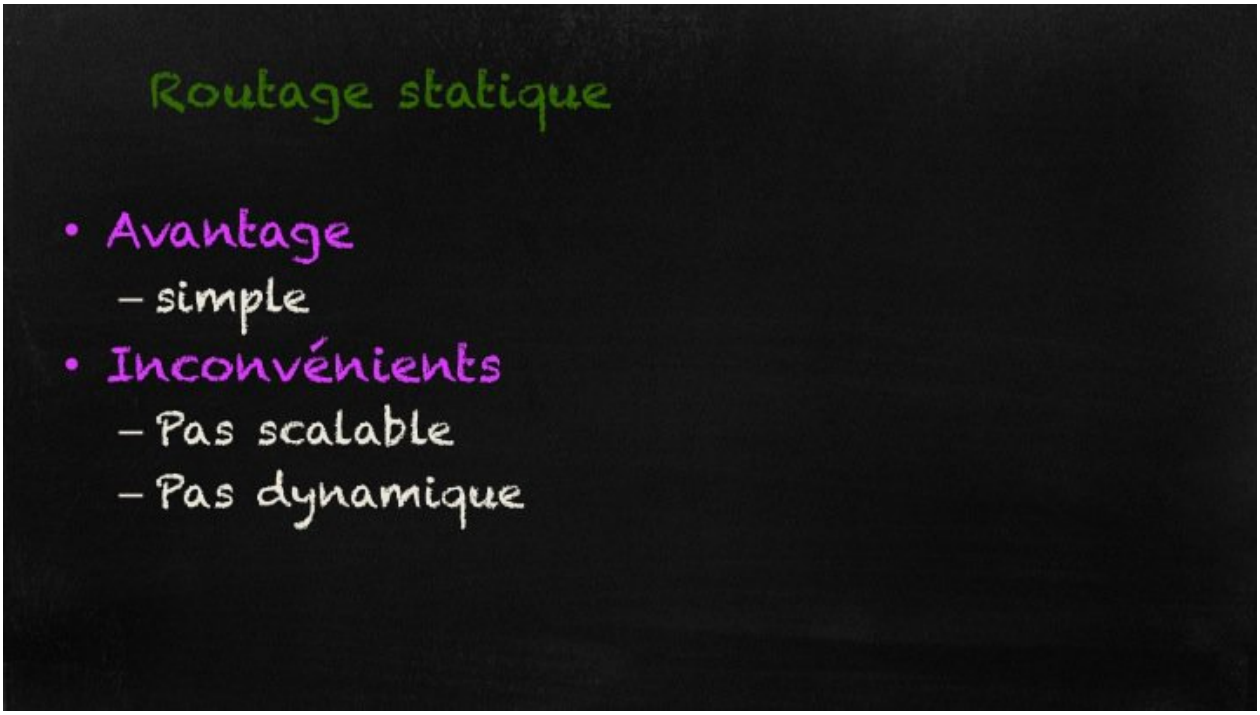


Le choix de ce critère revient à l'administrateur du réseau et les calculs sont effectués par le protocole de routage.
Ce critère permettra de donner une métrique (un coût) qui sera associée au chemin. Dans la table de routage, on gardera le chemin de plus petit coût.

Le routage statique



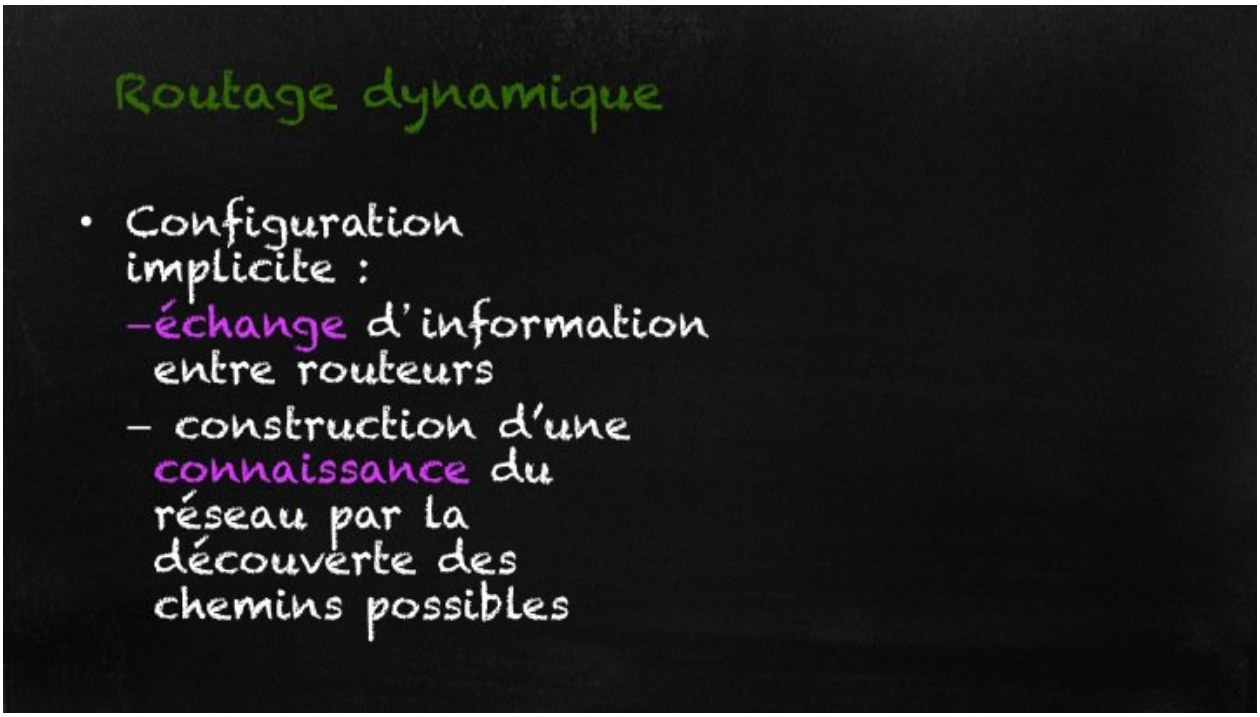
Une fois la route choisie, il faut rentrer les informations dans la table de routage. Une méthode simple serait d'utiliser du routage statique. L'administrateur configure alors explicitement les équipements et les routes utilisées dans les tables de routage de toutes les machines du réseau.



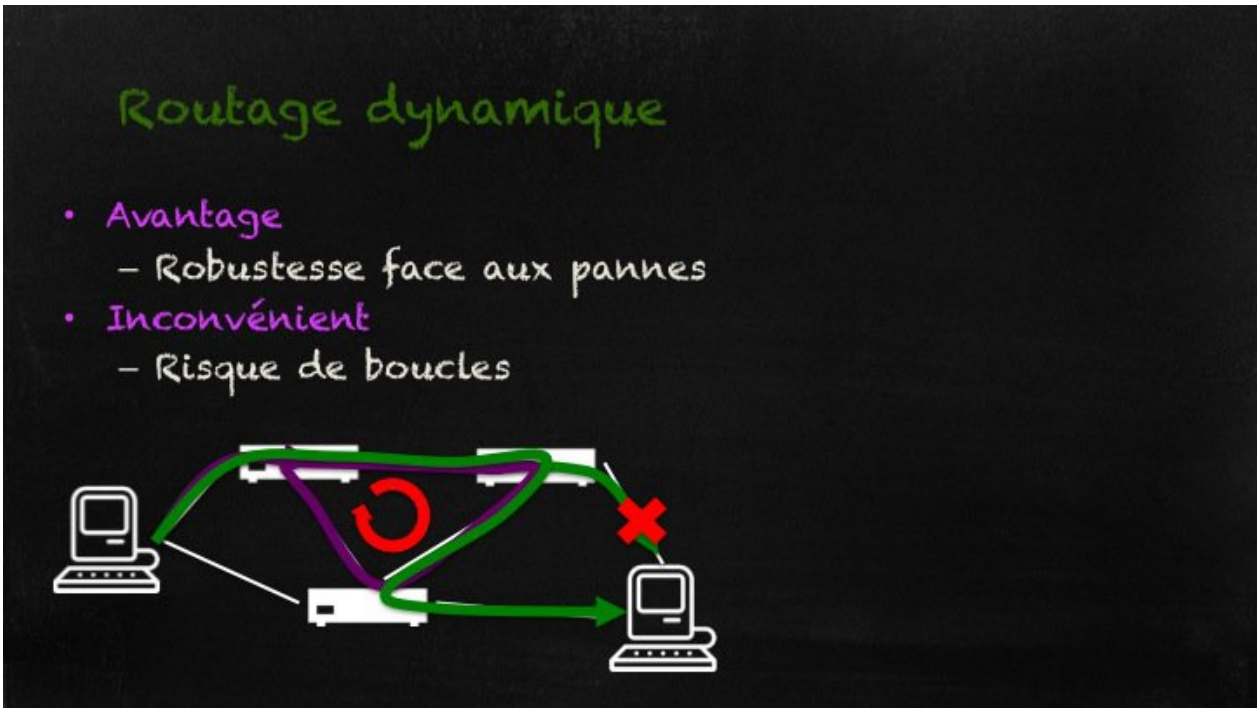
Cette méthode a l'avantage d'être simple et est suffisante pour un petit réseau. Mais elle pose des problèmes importants de passage à plus grande échelle : l'administrateur devra configurer toutes les machines du réseau.

De plus, cela ne prend pas en compte la « dynamicité » du réseau. Si un lien tombe (la topologie change), alors l'administrateur devra reconfigurer les tables de routage de toutes les machines du réseau pour prendre en compte la nouvelle route pour détourner le trafic.

Le routage dynamique



Les conditions dans le réseau changent. Des liens peuvent tomber et des équipements peuvent être en panne. Il faut prendre en compte ces événements lors que l'on construit les routes. Ainsi, on fera du routage dynamique. Les routeurs échangent des informations pour construire une connaissance commune du réseau de façon distribuée et découvrir les chemins possibles.



Le routage dynamique apporte une certaine robustesse face aux pannes. En revanche, il apporte un problème puisque, lorsqu'on va recalculer la route de détour, il faudra faire attention à ne pas créer de boucle. En effet, une boucle sur un chemin fait que le trafic n'arrive pas à destination. De plus, on utilise inutilement la bande passante sur ce chemin.

Quand on va chercher à recalculer la nouvelle route, le réseau se trouve dans un état instable. Pendant ce temps de convergence, il est possible que des boucles transitoires existent sur le réseau.

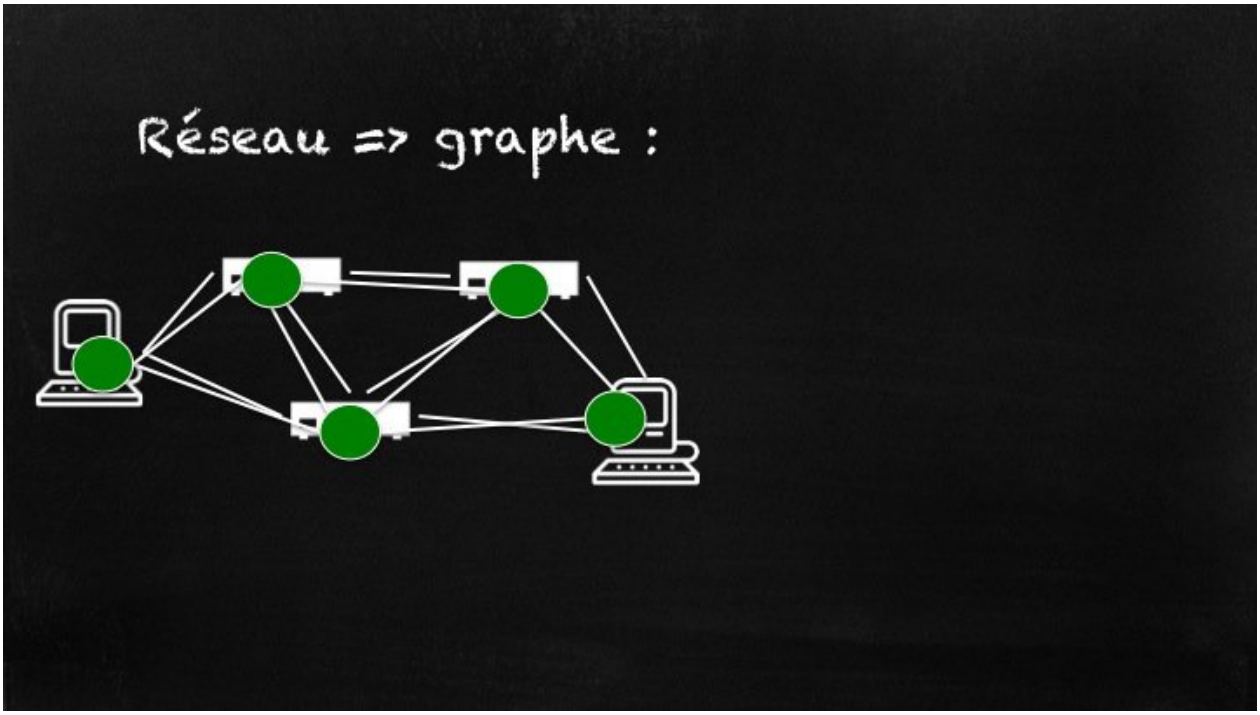
En résumé

Afin de construire des chemins sans boucle, on construira un arbre couvrant du réseau, aussi appelé arbre des plus courts chemins. Cet arbre répertorie les plus courts chemins entre une source et toutes les destinations atteignables dans le réseau. Dans la prochaine séance, nous verrons comment utiliser l'algorithme de Dijkstra pour construire cet arbre.

Séance 3 : Dijkstra

Introduction

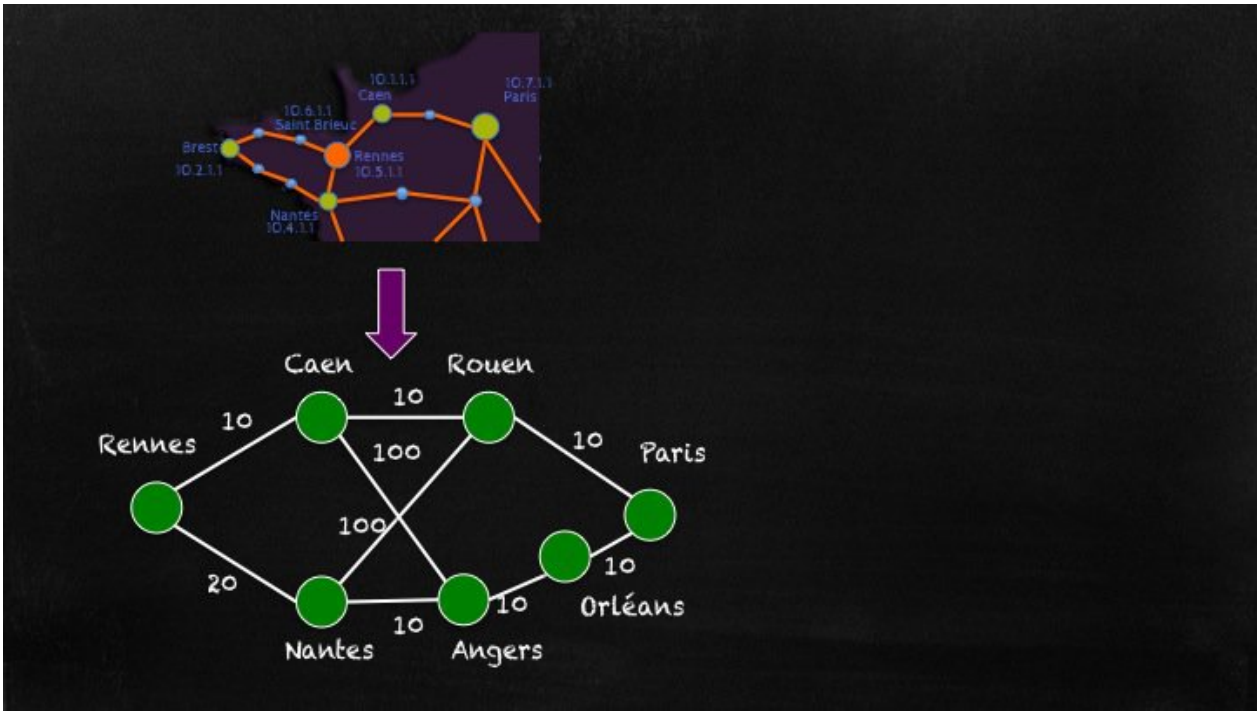
Le réseau réel est maillé : il existe un grand nombre de chemins possibles pour aller d'un point à un autre. L'algorithme de Dijkstra permet de construire, à partir du réseau réel, un arbre, dit couvrant car toutes les destinations sont couvertes / atteintes, des plus courts chemins dans le réseau, et assure que les chemins n'ont pas de boucle. Dans cette session, nous allons voir le principe de l'algorithme de Dijkstra.



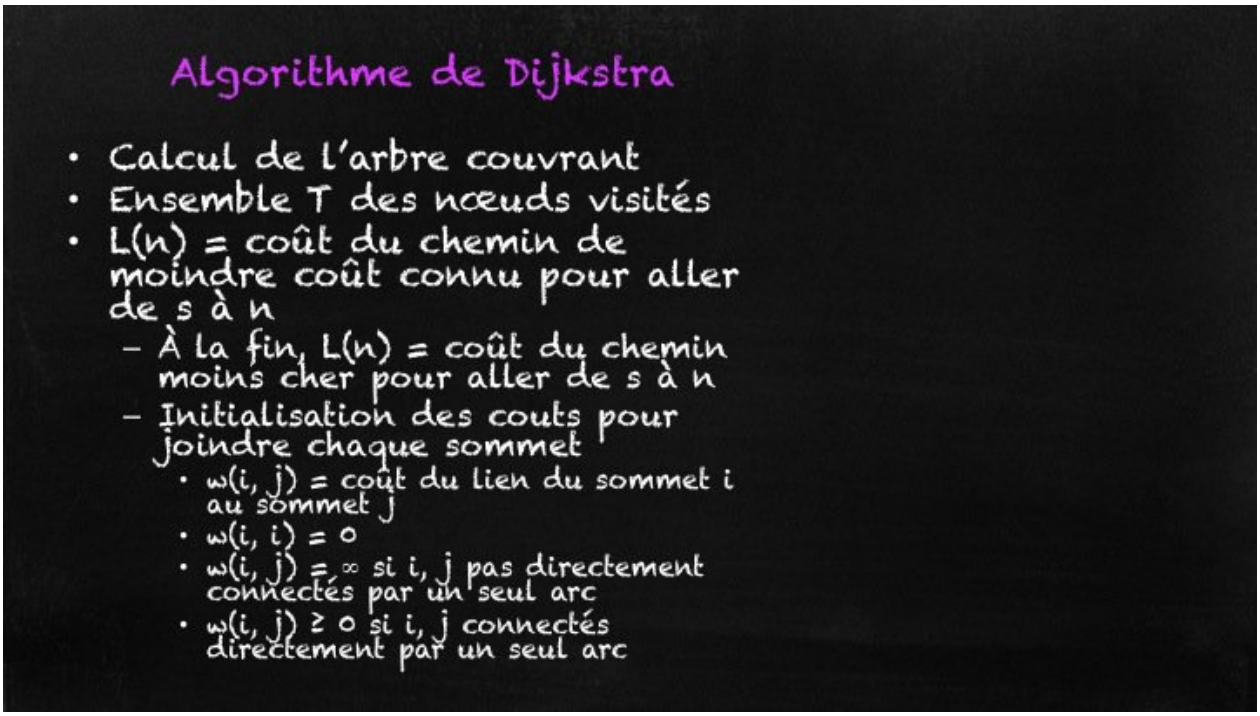
L'algorithme de Dijkstra est issu de la théorie des graphes. Pour l'appliquer, nous transformons la représentation du réseau en graphe maillé :

- Les équipements sont des sommets.
- Les liaisons sont des arcs.
- Les coûts associés aux liens sont les poids des arcs.

Nous connaissons donc pour chaque nœud ses voisins et le coût pour atteindre chaque voisin.



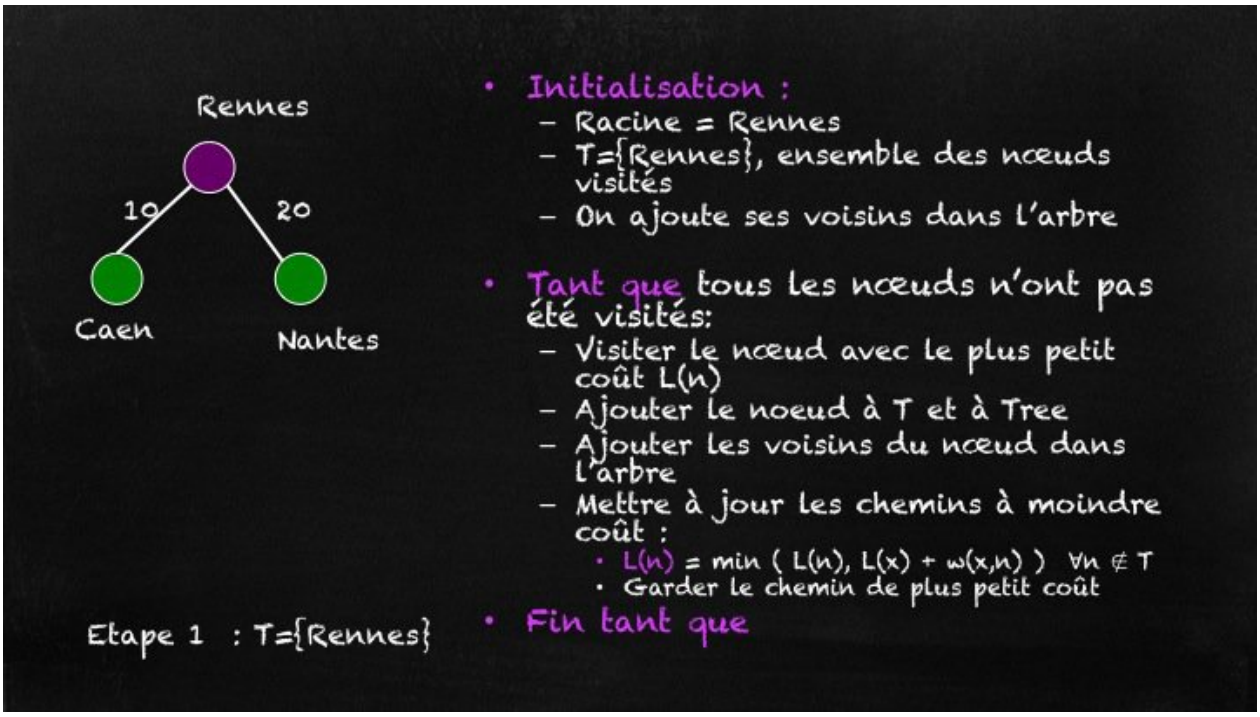
L'algorithme de Dijkstra



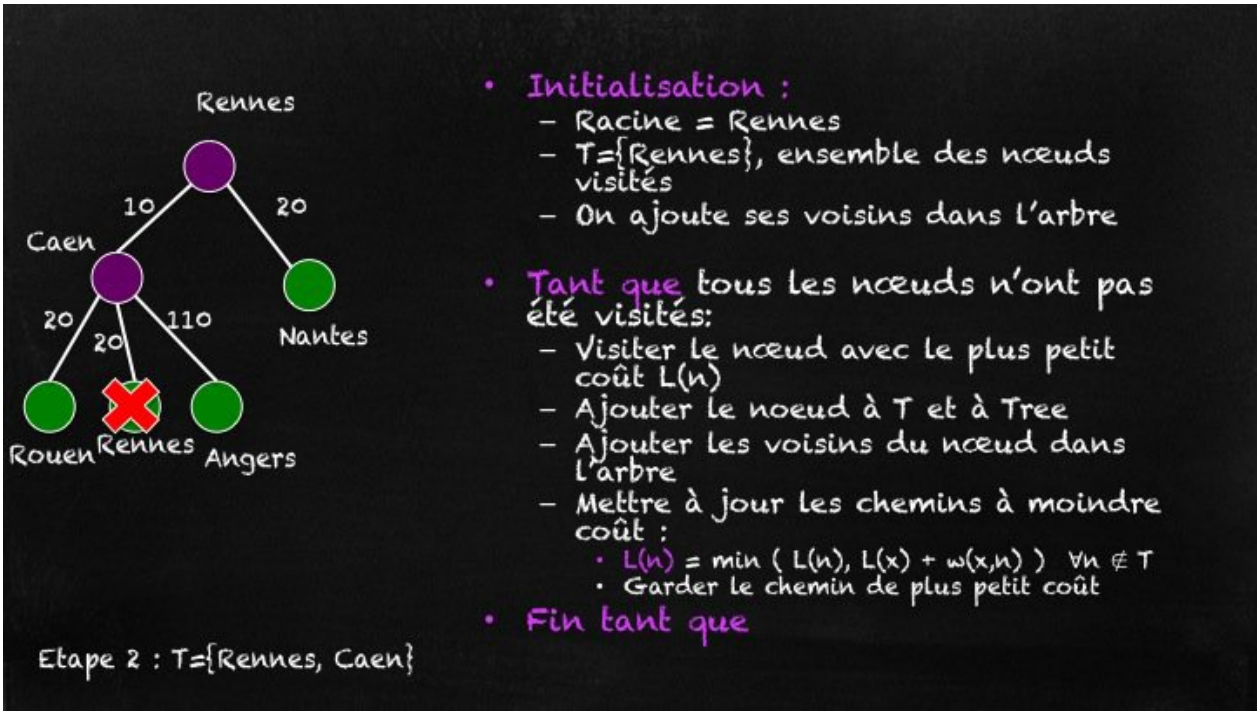
L'algorithme de Dijkstra construit l'arbre des plus courts chemins à partir du nœud courant. On va placer ce nœud à la racine de l'arbre. L'idée est ensuite d'explorer tous les sommets du graphe en commençant par la racine puis à chaque itération considérer un nouveau sommet.
A chaque itération, on regardera ce que ce nouveau sommet peut nous apprendre sur le graphe :

- quel nœud il nous permet d'atteindre ;
- à quel coût.

On ne gardera dans l'arbre que les chemins de plus petit coût.



L'algorithme va commencer la construction de l'arbre en insérant à la racine le nœud courant (Rennes). En effet, nous cherchons les chemins de plus petits coûts de ce nœud vers tous les nœuds du réseau. Nous voulons construire l'arbre des chemins de plus petit coût de la racine vers toutes les destinations du réseau.
Ensuite, nous insérons tous les voisins de Rennes en indiquant le coût pour les atteindre : Caen avec un coût de 10 et Nantes avec un coût de 20. Pour l'étape suivante, nous allons choisir un de ces sommets et explorer ce que l'on peut apprendre à partir de celui-ci. Afin de converger plus vite vers la solution, nous choisirons le sommet de plus petit coût. Ici, ce sera Caen.

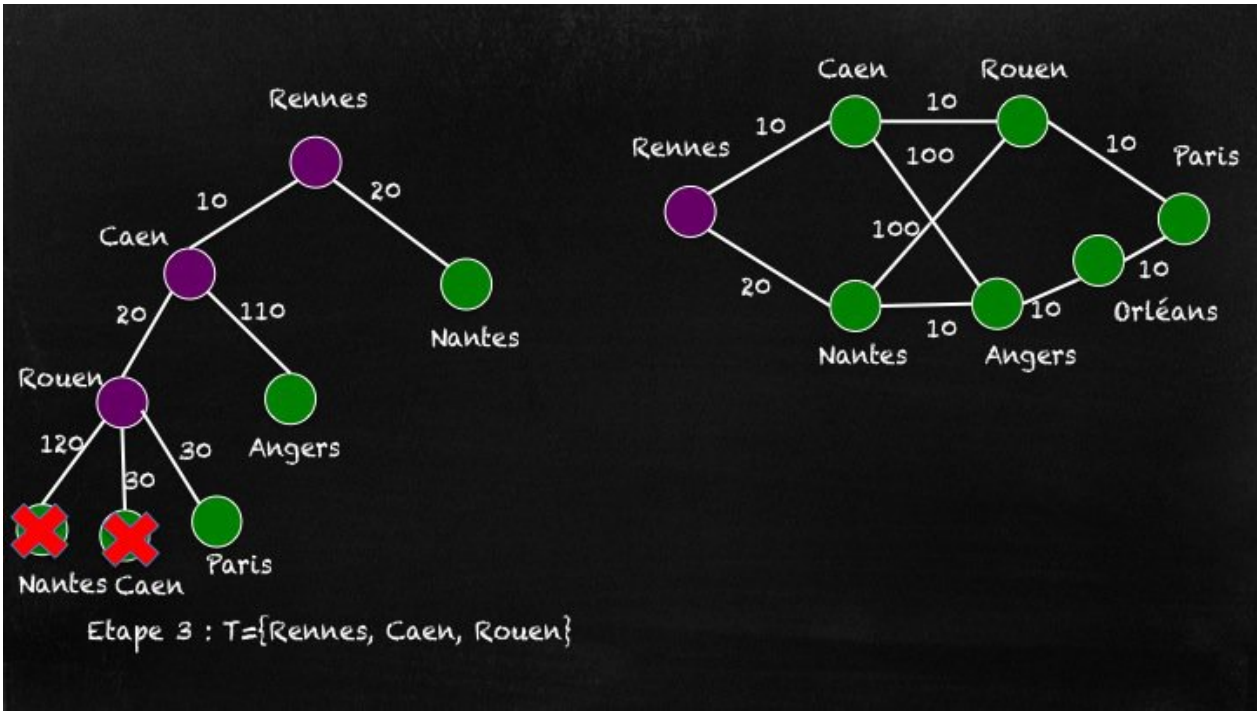


A partir de Caen, nous pouvons atteindre Rouen, Rennes et Angers :

- Rouen avec un coût de 20 car pour aller à Caen nous avions déjà un coût de 10, et nous ajoutons l'arc Caen – Rouen qui a un coût de 10 également ;
- Angers avec un coût de 110 ;
- Rennes avec un coût de 20 ; cette branche ne sera pas gardée car pour aller de Rennes à Rennes, il n'est pas utile de passer par Caen.

Nous venons de considérer le nœud de Caen. Nous l'insérons dans T, liste des sommets du graphe déjà visités. À cette itération, T contient Rennes et Caen.

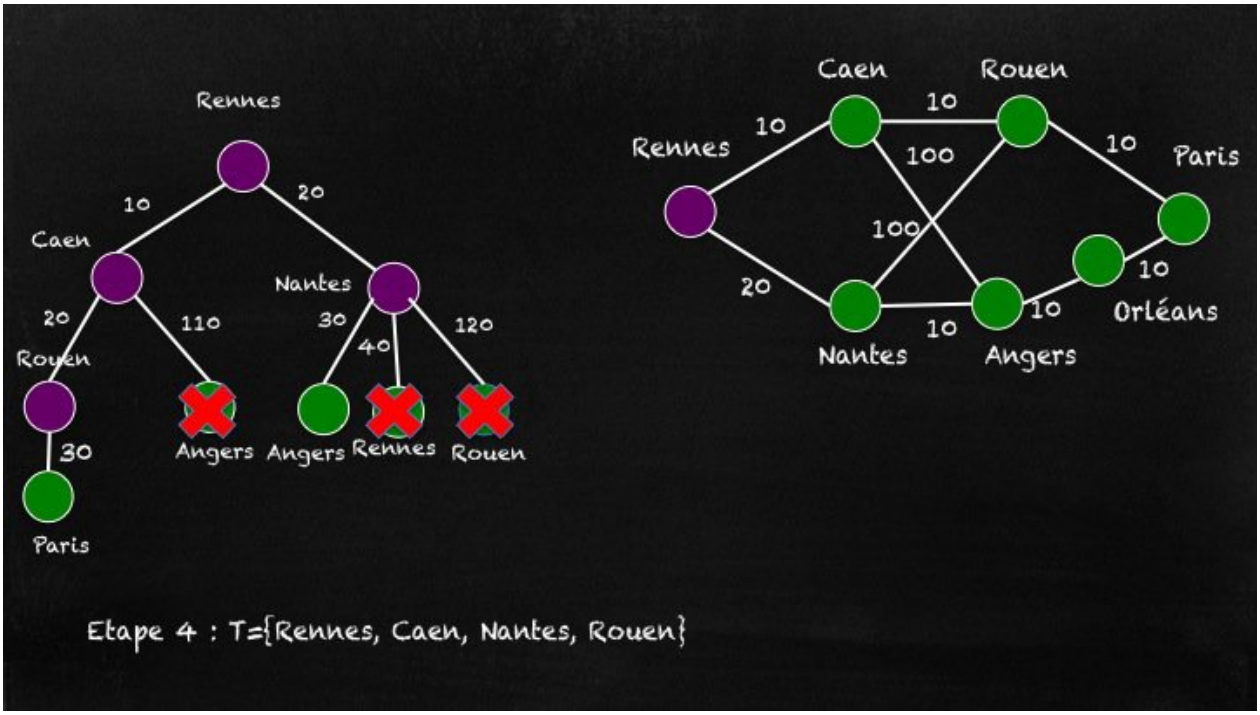
Nous allons pouvoir passer à l'itération suivante et choisir le nœud de plus petit coût non encore exploré. Nous avons le choix entre Rouen et Nantes qui sont de coût égal. Il faut en choisir un : choisissons Rouen.



Rouen a pour voisins Nantes, Caen et Paris :

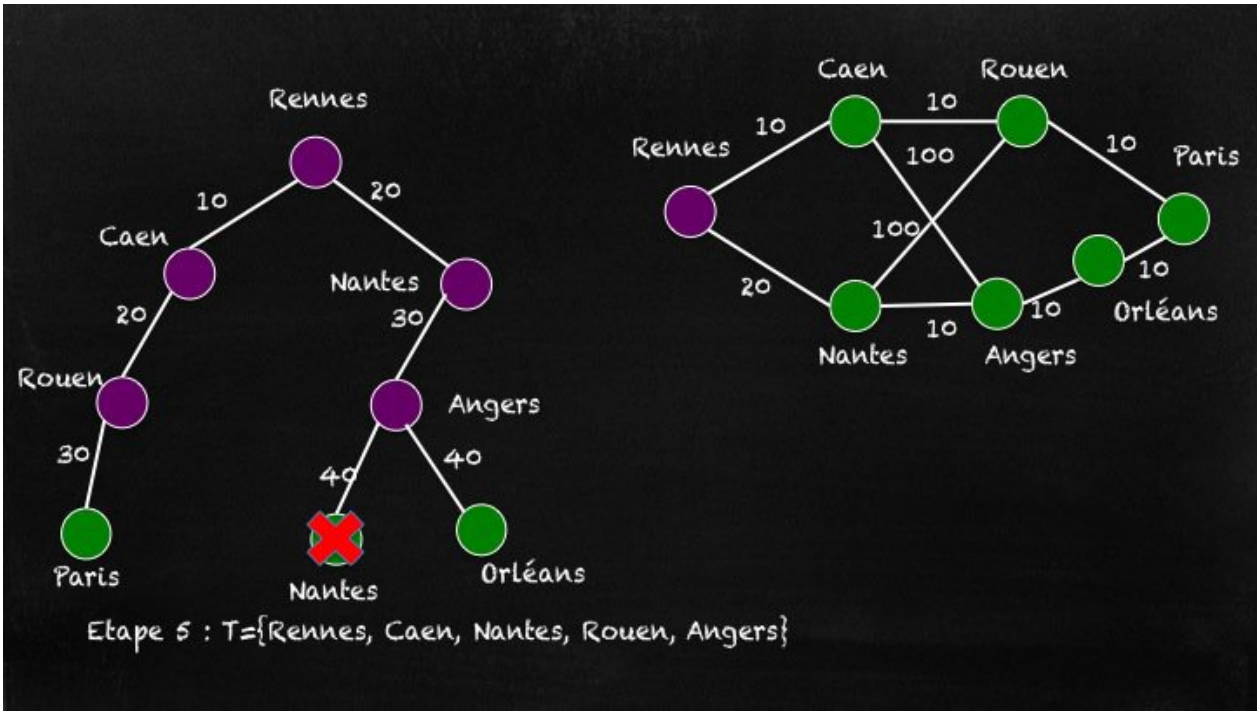
- Nantes aura un coût de 120 ; ce qui est plus grand que le coût du chemin que nous avions déjà puisqu'il possède un coût de 20. Cette branche ne sera pas gardée.
- Caen aura un coût de 30 alors que nous avions déjà une route avec un coût de 10. Cette branche ne sera pas gardée non plus.
- Paris aura un coût de 30, ce qui améliore notre connaissance puisque Paris n'avait pas encore été découvert dans l'arbre.

Nous ajoutons Rouen à l'ensemble T des sommets visités et nous passons à l'exploration du sommet suivant : Nantes.



Nantes a pour voisins : Angers (avec un coût de 30), Rennes (avec un coût de 40) et Rouen (avec un coût de 120). Les branches vers Rennes et Rouen n'apportent pas de meilleures connaissances et ne seront pas gardées. En revanche, le chemin pour aller de Rennes à Angers avec un coût de 30 en passant par Nantes est plus intéressant que le chemin que nous avions en passant par Caen et qui avait un coût de 110. Nous allons donc la garder et détruire dans l'arbre le chemin qui allait de Rennes à Angers en passant par Caen.

Nous ajoutons Nantes à l'ensemble T des sommets visités et nous passons à l'exploration du sommet suivant. Nous avons le choix entre deux chemins de plus petits coûts et équivalents : « Rennes – Angers » ou « Rennes – Paris ». Prenons Angers.



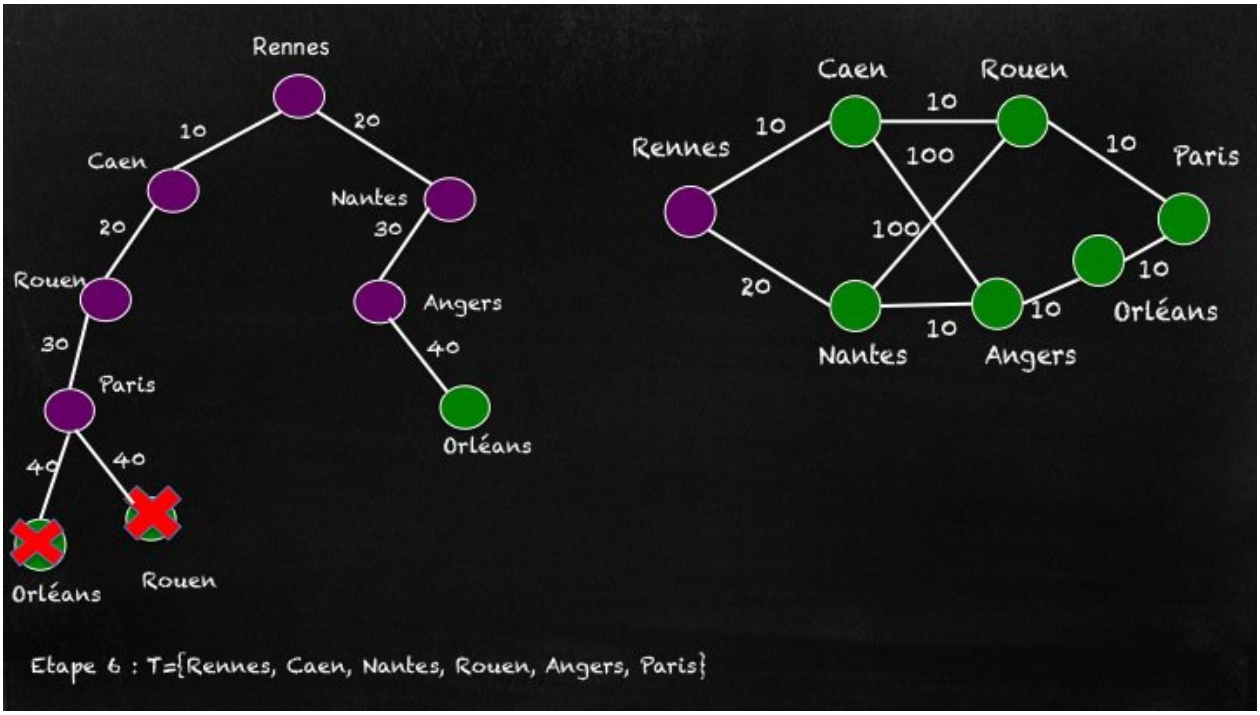
A partir d'Angers, nous découvrons les deux sommets voisins suivants :

- Orléans avec un coût de 40 ;
- Nantes avec un coût de 40.

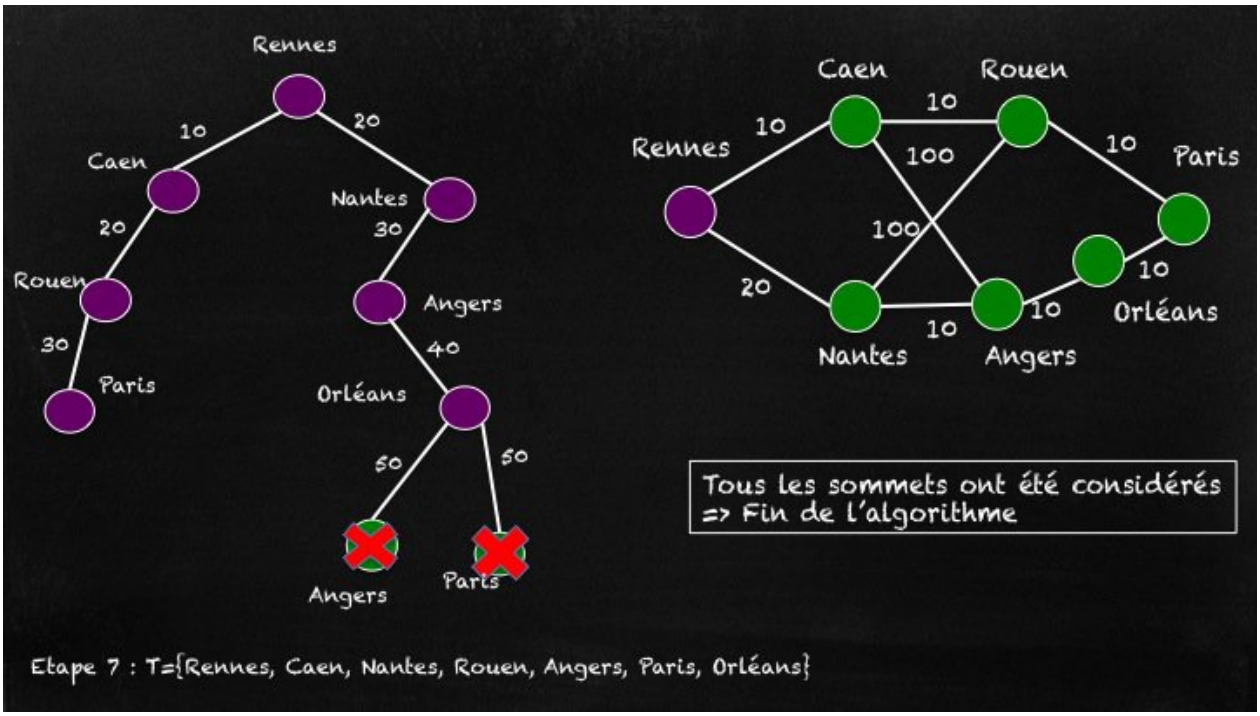
Le chemin pour Nantes ne nous intéresse pas, il n'apportent pas de meilleure connaissance et ne sera pas gardé. En revanche, le chemin pour aller de Rennes à Orléans avec un coût de 40 est intéressant car Orléans ne figurait pas dans l'arbre.

Vous noterez au passage que deux chemins différents peuvent avoir le même coût. Le chemin allant de Rennes à Orléans passant par Rouen et Paris, bien qu'ayant le même coût que celui passant par Nantes et Angers, est abandonné car trouvé en second. Le choix entre deux chemins de même coût est donc arbitraire.

Nous ajoutons Angers à l'ensemble T des sommets visités et nous passons à l'exploration du sommet suivant : Paris.

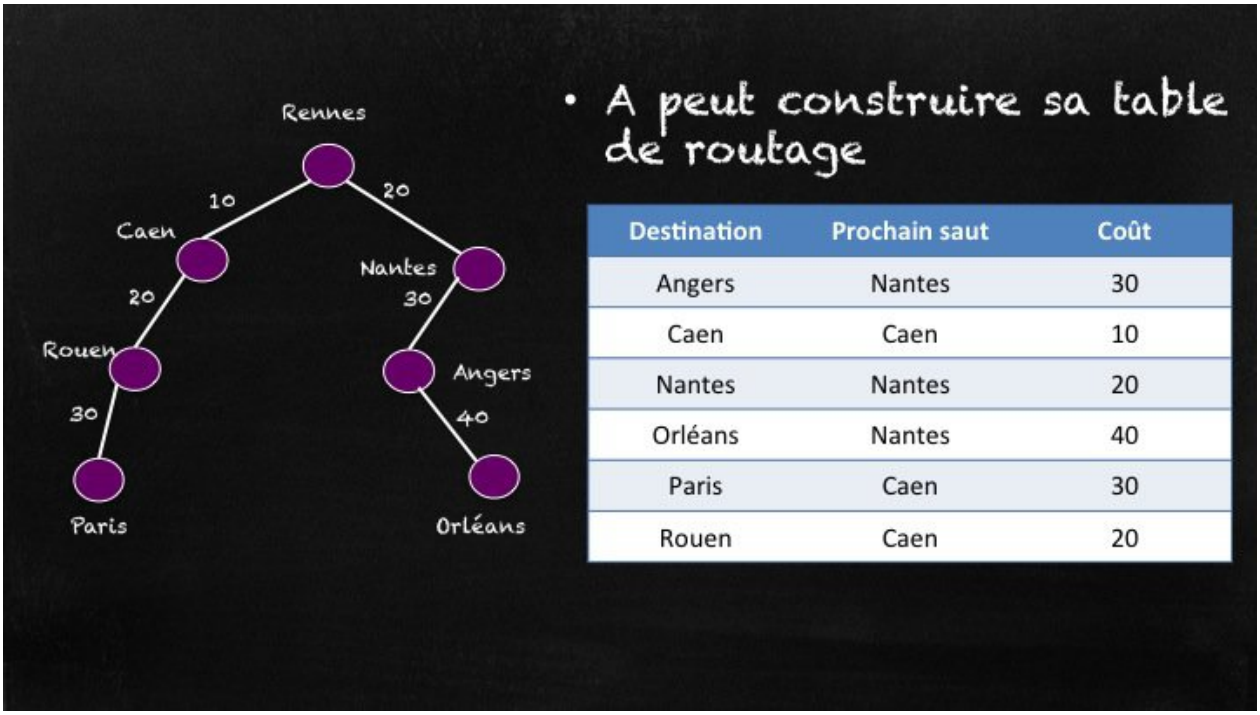


Nous insérons deux branches supplémentaires : Orléans avec un coût de 40 et Rouen avec un coût de 40. Les coûts sont supérieurs à la connaissance que nous avons déjà : Paris ne nous apprend rien de nouveau.
Nous ajoutons Paris à l'ensemble T des sommets visités et nous passons à l'exploration du sommet suivant : Orléans.



Orléans a pour voisins Angers et Paris mais ces branches n'améliorent pas notre connaissance et ne seront pas gardées.

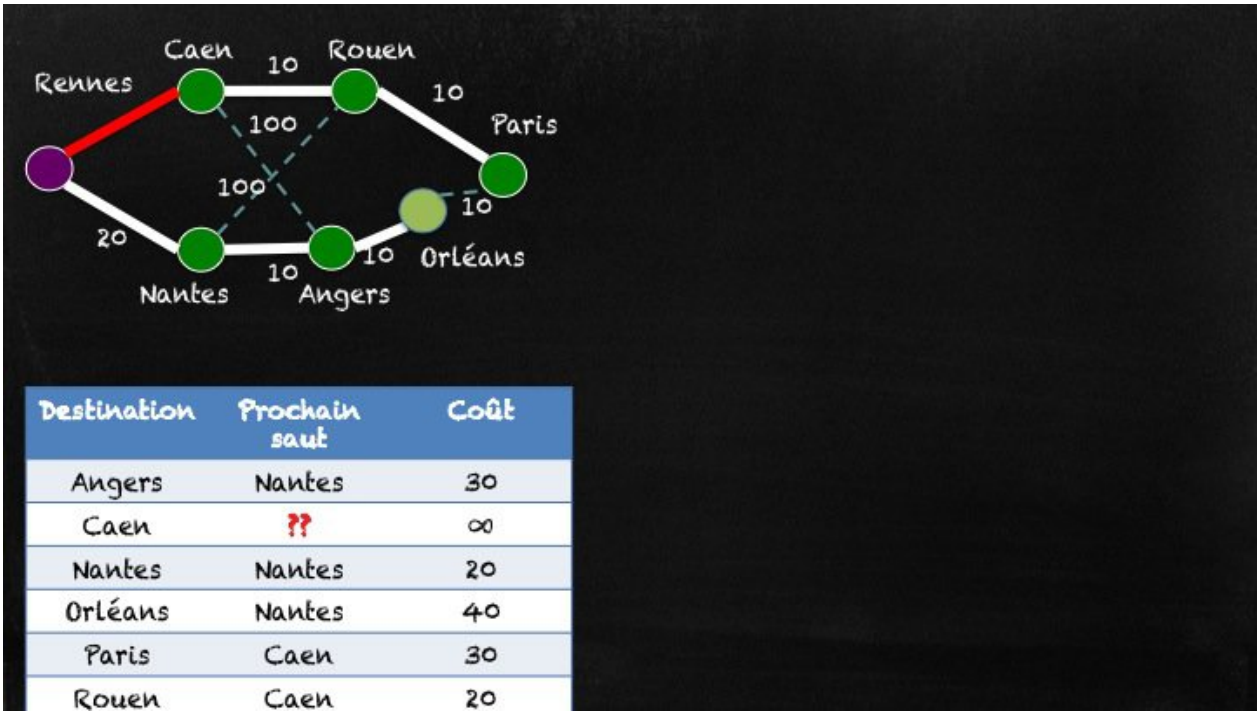
Nous ajoutons Orléans à l'ensemble T des sommets visités. Nous avons visité tous les sommets du graphe et avons donc terminé les itérations.



Destination	Prochain saut	Coût
Angers	Nantes	30
Caen	Caen	10
Nantes	Nantes	20
Orléans	Nantes	40
Paris	Caen	30
Rouen	Caen	20

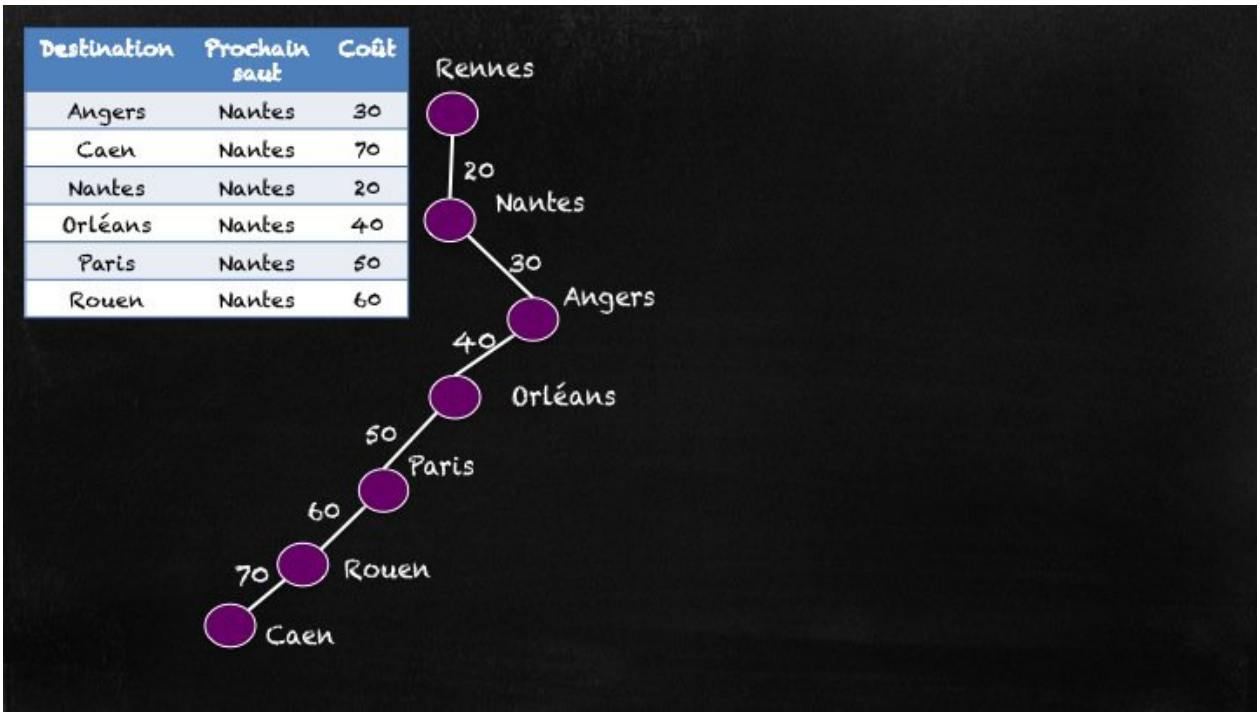
Une fois l'arbre des plus courts chemins calculé, la racine, Rennes, peut mettre sa table de routage à jour.

Prise en compte des pannes



Destination	Prochain saut	Coût
Angers	Nantes	30
Caen	??	∞
Nantes	Nantes	20
Orléans	Nantes	40
Paris	Caen	30
Rouen	Caen	20

En cas de panne d'un lien ou d'un équipement, de nombreuses routes ne seront plus valides. Il faudra alors recalculer un nouvel arbre couvrant pour trouver des chemins de secours.



Imaginons que la panne survienne entre Rennes et Caen. Les nœuds vont alors exécuter à nouveau l'algorithme de Dijkstra pour trouver un nouvel arbre couvrant. Le chemin le plus long dans l'arbre permet d'atteindre Caen avec un coût de 70. Rennes peut alors mettre à jour sa table de routage et, pour toutes les destinations, le prochain saut sera Nantes.

En résumé

Dans cette séance, nous avons vu l'algorithme de Dijkstra qui permet de calculer l'arbre des plus courts chemins d'un nœud vers tous les autres nœuds du réseau. Cet arbre est utilisé pour renseigner la table de routage. L'un des principaux avantages de l'algorithme de Dijkstra est qu'il a une terminaison en un temps polynomial. En effet, il se termine quand tous les nœuds ont été examinés. Son principal problème, pour son utilisation dans un réseau, est qu'il suppose que chaque nœud du réseau ait une connaissance complète de la topologie du réseau.

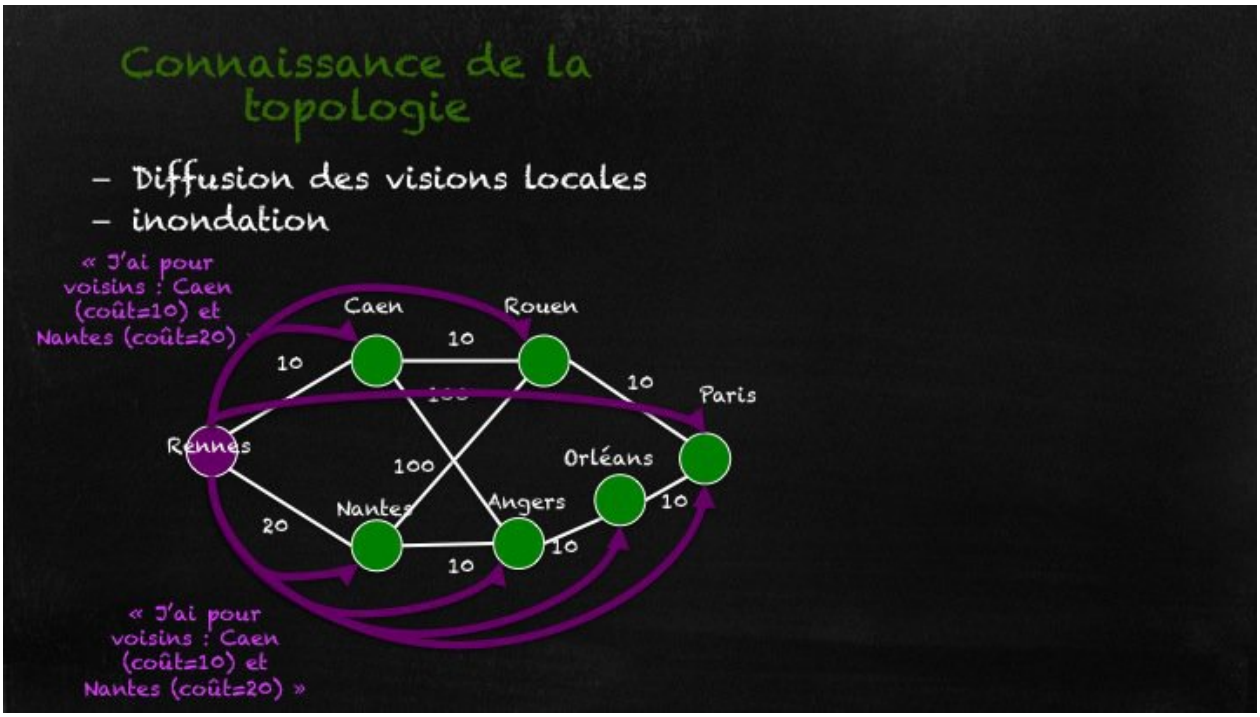
Séance 4 : OSPF

Introduction

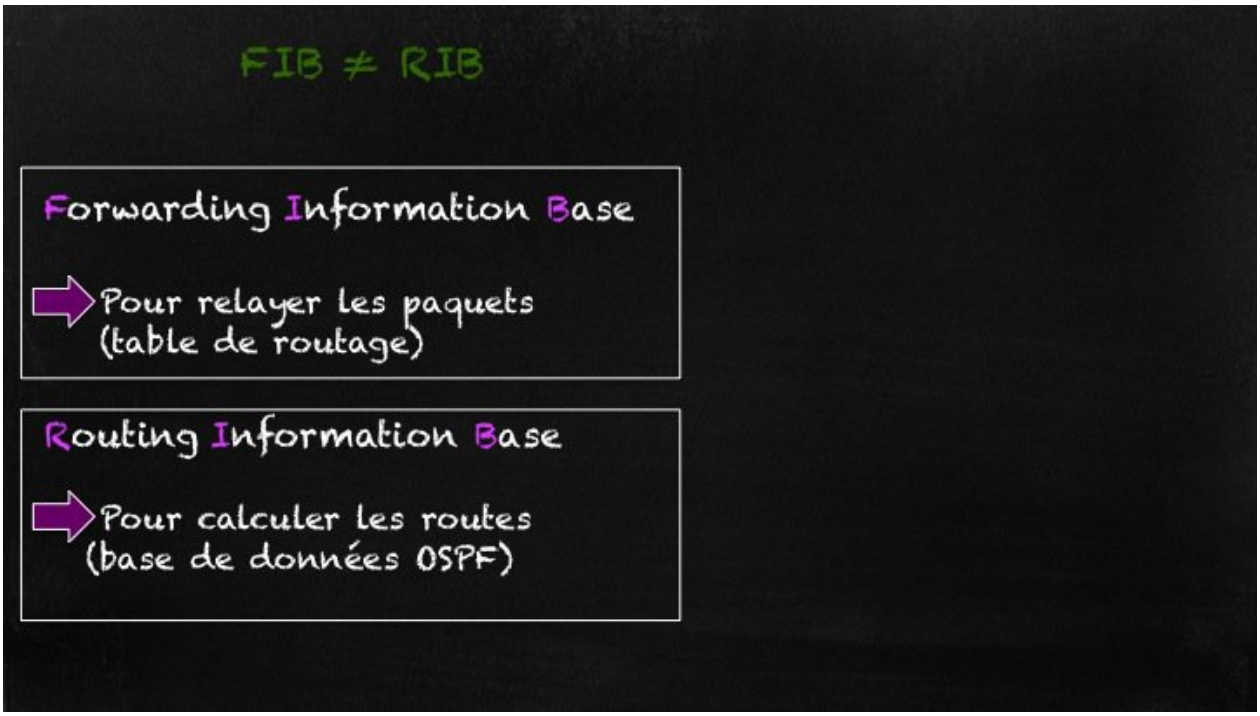
L'algorithme de Dijkstra que nous avons vu est utilisé de façon distribuée par chaque routeur pour sélectionner les prochains sauts à garder dans la table de routage. Pour s'exécuter, cet algorithme doit connaître la topologie du réseau. Les protocoles de routage (RIP, OSPF, ISIS…) proposent, en plus d'un algorithme de sélection du prochain saut, des formats de messages et des mécanismes pour échanger des informations et découvrir cette topologie.
Dans cette séance, nous allons aborder l'un d'entre eux : le protocole OSPF (Open Shortest Path First) qui met en œuvre le routage par le plus court chemin disponible vers la destination.

OSPF est un protocole de routage standardisé à l'IETF. Il s'adresse à des **réseaux maillés** de grande taille (quelques dizaines voire centaines de nœuds).
Comme tout protocole de routage distribué, il s'exécute sur chaque routeur du réseau et suppose un dialogue entre les routeurs. Ce trafic de signalisation, transporté au dessus d'IP reste faible.
Ce protocole est de type "état de liens" *link state* ; c'est-à-dire que chaque routeur dispose d'une vision complète et exacte de la topologie du réseau.

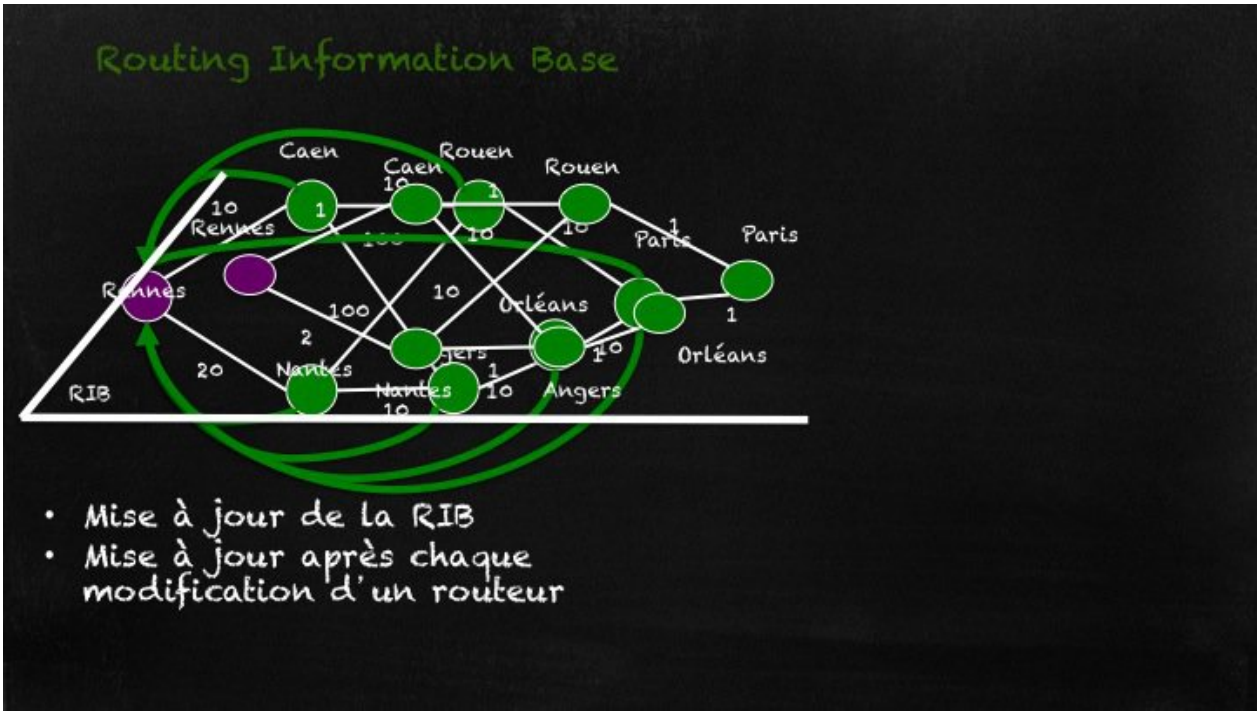
Vision de la topologie



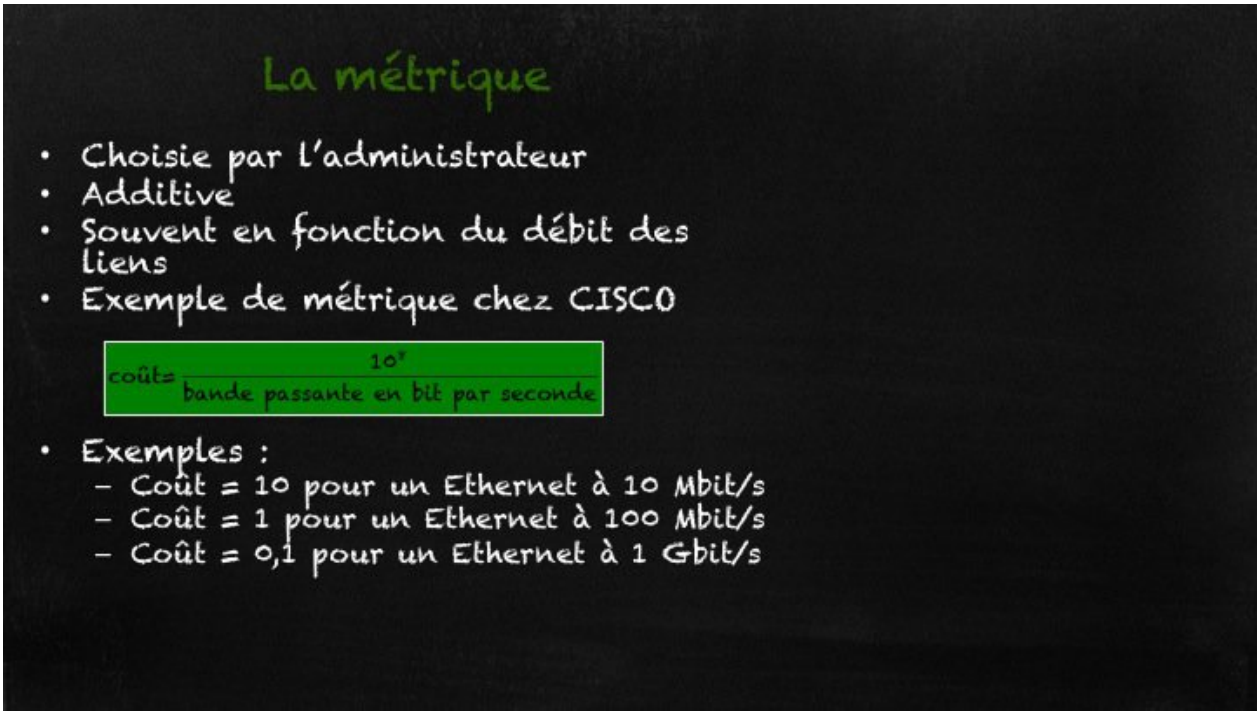
OSPF s'appuie sur la connaissance de la topologie par chaque nœud. Pour construire cette vision localement, chaque routeur va diffuser à l'ensemble du réseau la liste de ses voisins et le coût pour les atteindre. Ceci est fait par un mécanisme d'inondation.



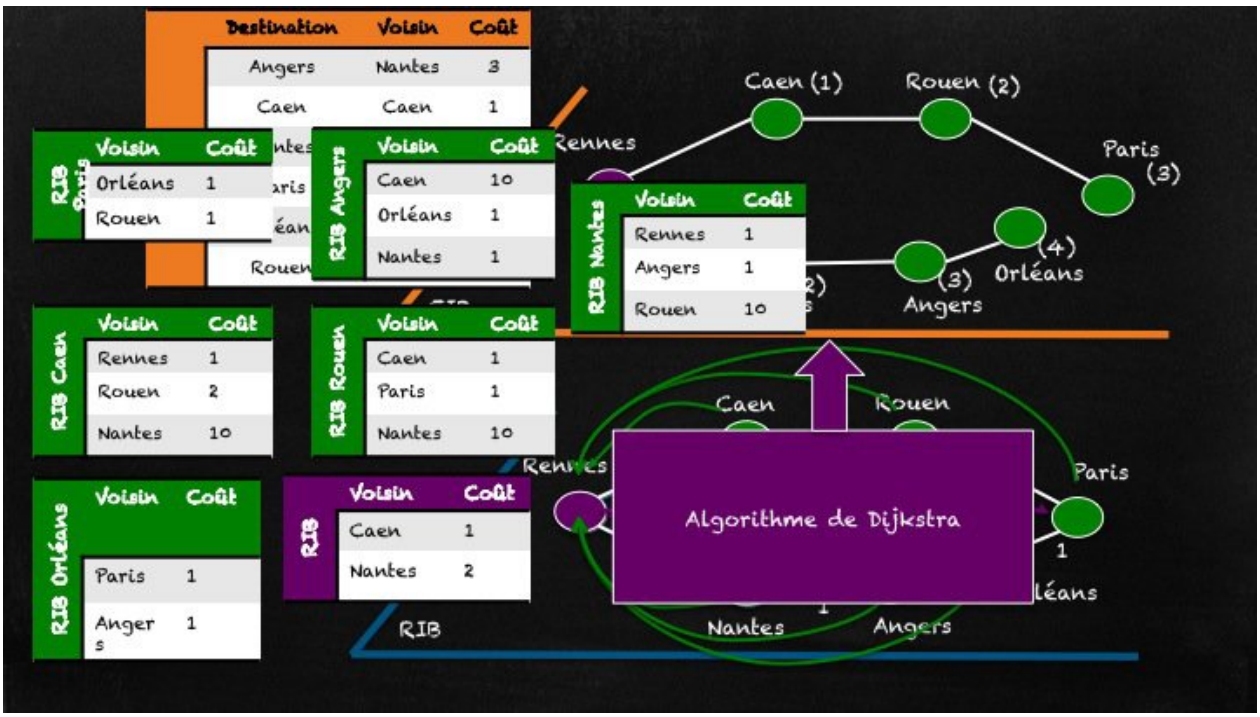
Nous avons vu que le terme de table de routage est ambigu. Dans OSPF, on différencie la table servant au relayage des paquets : la FIB (*Forwarding Information Base)*, de la table qui permet de calculer et de choisir les chemins : la RIB (*Routing Information Base)*, aussi appelée base de donnée OSPF, qui contient la topologie.



Chaque routeur reçoit les relations d'adjacence dans le réseau et peut construire sa vision de la topologie, localement, qui sera stockée dans la RIB.
Cette vision sera utilisée pour exécuter l'algorithme de Dijkstra et mettre à jour la table pour le relayage des paquets (FIB).
Un changement dans la topologie peut être vite signalé par les routeurs et pris en compte en recommençant ces opérations.

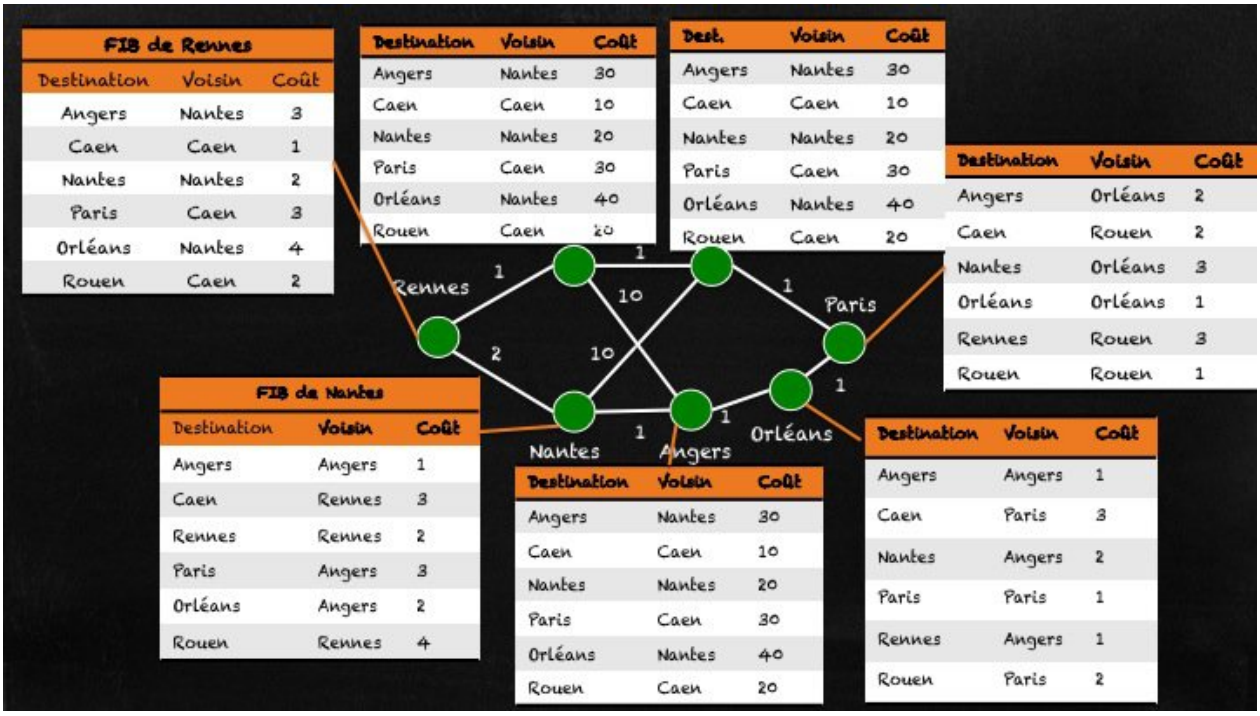


Les coûts associés aux liaisons dépendent de la métrique choisie par l'administrateur réseau. OSPF utilise une métrique additive qui fait souvent intervenir le débit des liens. Par exemple, CISCO calcule un coût faible quand le débit de l'interface est grand. Cela permet de préférer les chemins proposant le plus de bande passante.

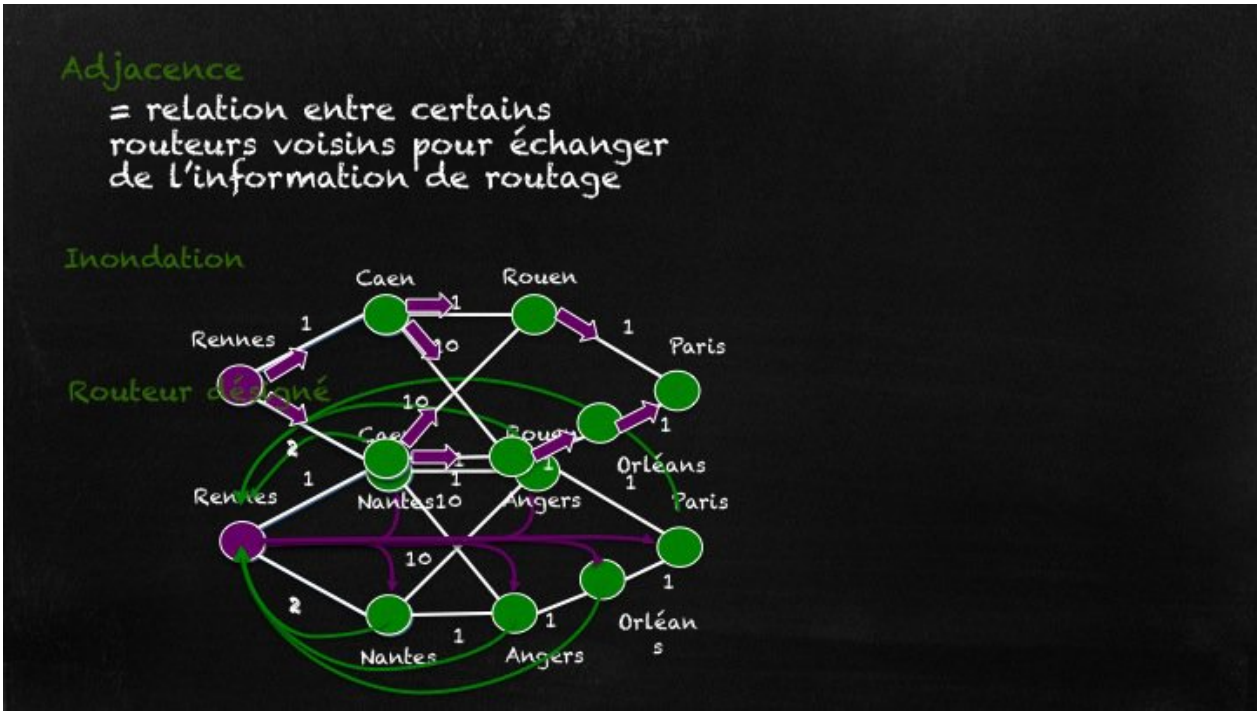


Reprenons notre exemple pour dérouler le protocole :

- Le routeur de Rennes est connecté à deux voisins : Caen et Nantes, avec des liaisons ayant un coût respectif de 1 et 2. Cette information est fiable. Il la diffuse aux autres routeurs du réseau.
- Tous les routeurs font de même.
- Rennes reçoit donc leur information topologique.
- Le routeur de Rennes peut donc construire sa vision de la topologie et la stocker dans sa RIB.
- Rennes exécute l'algorithme de Dijkstra et met à jour sa FIB que l'on appelle communément en français la table de routage.
- Les autres routeurs exécutent la même suite d'actions en parallèle => à la fin, mais pas forcément au même instant, tous les nœuds ont des FIB à jour et peuvent relayer correctement les paquets.



Les messages échangés



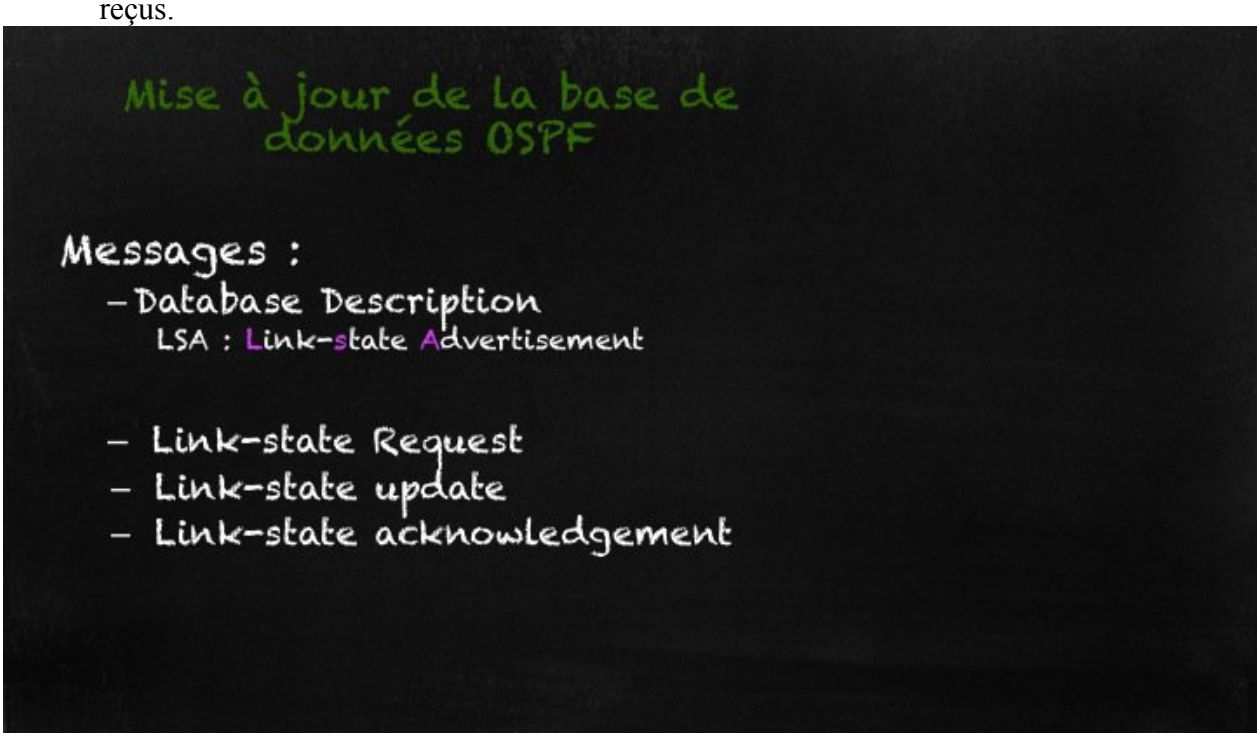
OSPF définit la notion d'adjacence pour désigner la relation entre deux routeurs qui échangent les informations nécessaires pour faire fonctionner OSPF (la topologie). Le procédé utilisé pour diffuser les informations à l'ensemble du réseau est appelé inondation et ne sera pas détaillé ici car il dépend de la nature du réseau. Plusieurs mécanismes peuvent être utilisés. OSPF définit la notion de routeur désigné. Ce routeur est élu dans le réseau pour maintenir et diffuser la liste des liens du réseau. Il est adjacent à tous les autres nœuds et peut permettre l'inondation.

OSPF est mis en œuvre par l'échange d'un ensemble de messages. On peut distinguer deux types de messages : ceux qui permettent de maintenir l'adjacence, et ceux qui maintiennent la synchronisation de la base de données OSPF dans les différents routeurs.

La découverte des voisins et le maintien de l'adjacence sont mis en œuvre par l'envoi périodique d'un message Hello. L'envoi d'un message Hello vers un voisin vérifie que la communication est possible vers ce voisin. La réception d'un message Hello envoyé par le voisin montre que la liaison permet une communication bidirectionnelle.

Le message Hello est court et contient :

- la liste des voisins déjà découverts,
- l'identité du routeur désigné et de son remplaçant en cas de panne,
- l'intervalle utilisé dans le réseau entre l'envoi de deux messages Hello,
- le délai au bout duquel la liaison sera déclarée en panne si les messages Hello ne sont plus reçus.



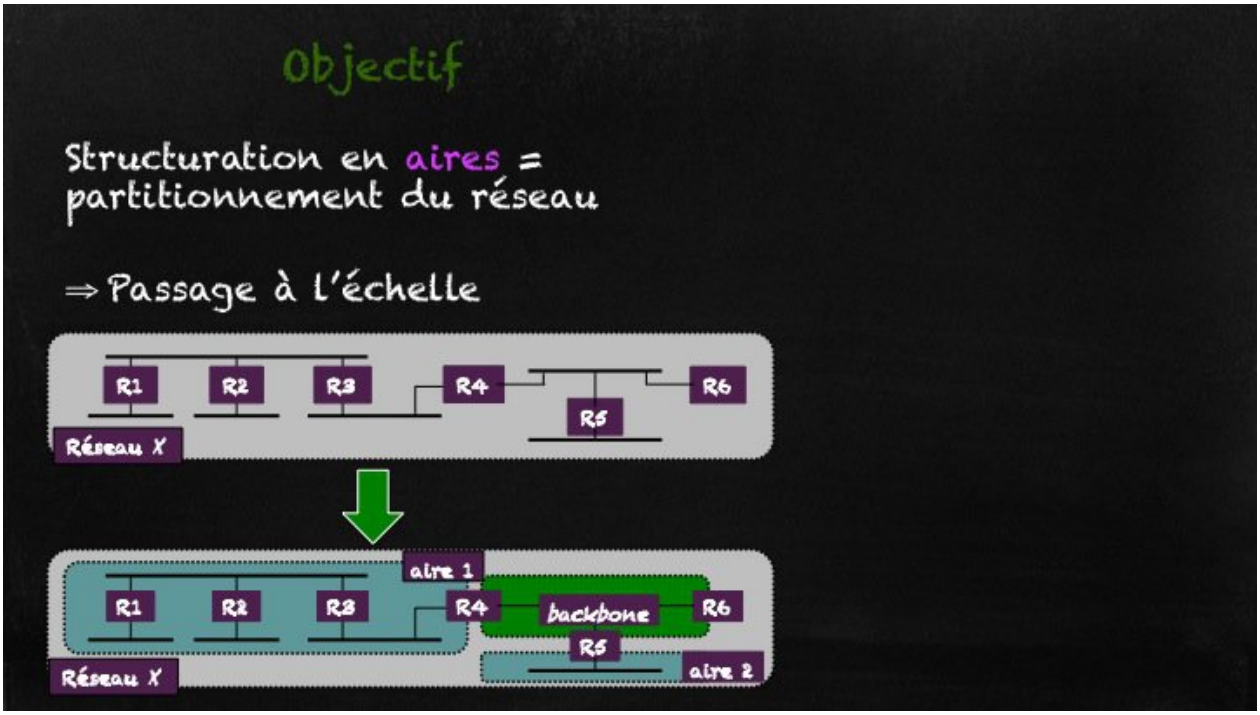
Une fois l'adjacence établie, les informations sur la topologie peuvent être échangées grâce à des paquets de description des bases de données. Ces paquets contiennent la liste des annonces des états de liens du réseau (LSA *Link State Advertisement*).
Plusieurs types de LSA existent.

- Une durée de vie est associée à ces états de liens. Les messages *LinkState Request* sont utilisés pour mettre à jour un état de lien trop vieux dans la base de données.
- Les messages *LinkState Update* et *Link State Acknowledgement* sont utilisés pour l'inondation ; c'est-à-dire la diffusion fiable des annonces d'états de liens.

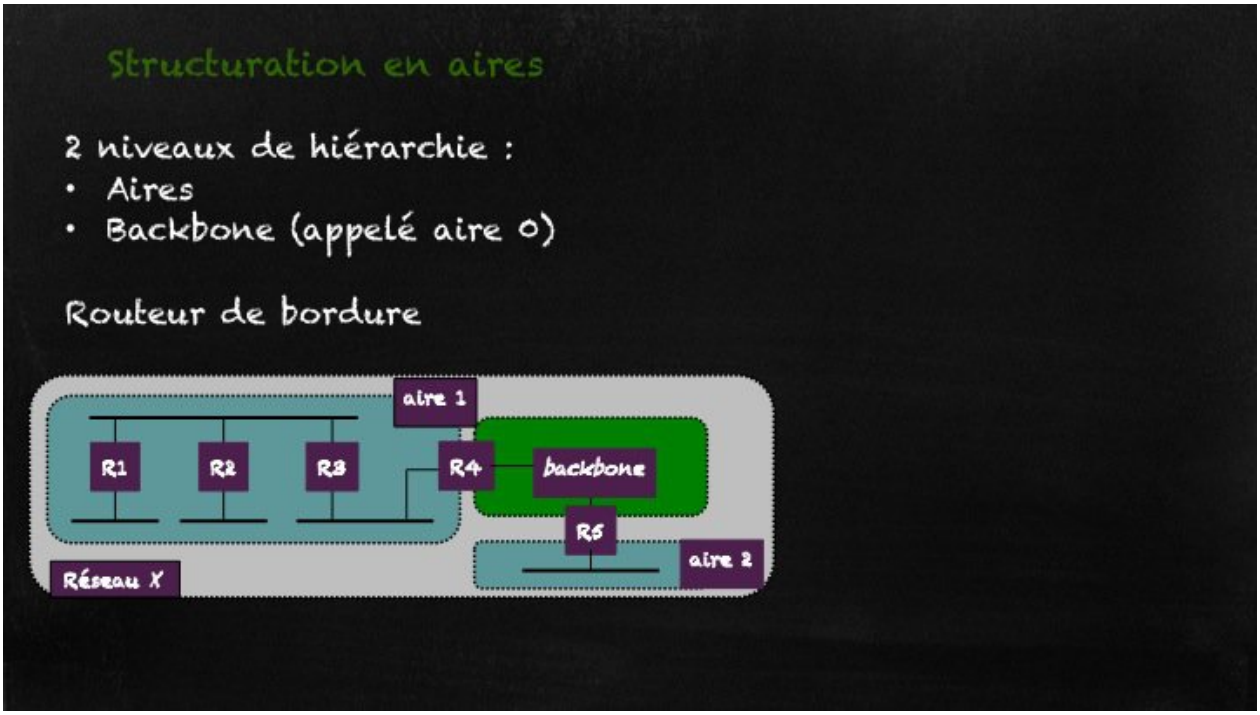
Comme tous les protocole de routage, OSPF nécessite des compromis :

- le premier compromis est entre la stabilité des routes et la réactivité aux pannes :
    - Ne pas réagir suffisamment vite à une panne de lien entraine la perte du trafic. En réagissant, OSPF calcule un chemin alternatif et détourne le trafic.
    - S'il réagit dès qu'il apprend la panne d'un lien ou d'un équipement et si l'équipement n'est pas vraiment en panne mais a une interface vacillante, OSPF va générer beaucoup d'instabilité dans les routes ; ce que certaines applications ne supportent pas.
- le second compromis est entre la réactivité du protocole et la quantité d'informations échangées.
    - Si les routeurs adjacents échangent souvent des informations, ils auront une vision très à jour du réseau et pourront prendre de bonnes décisions de routage...
    - mais auront généré beaucoup de trafic ; ce qui consomme de la bande passante qui ne pourra pas être utilisée pour les flux utilisateurs.

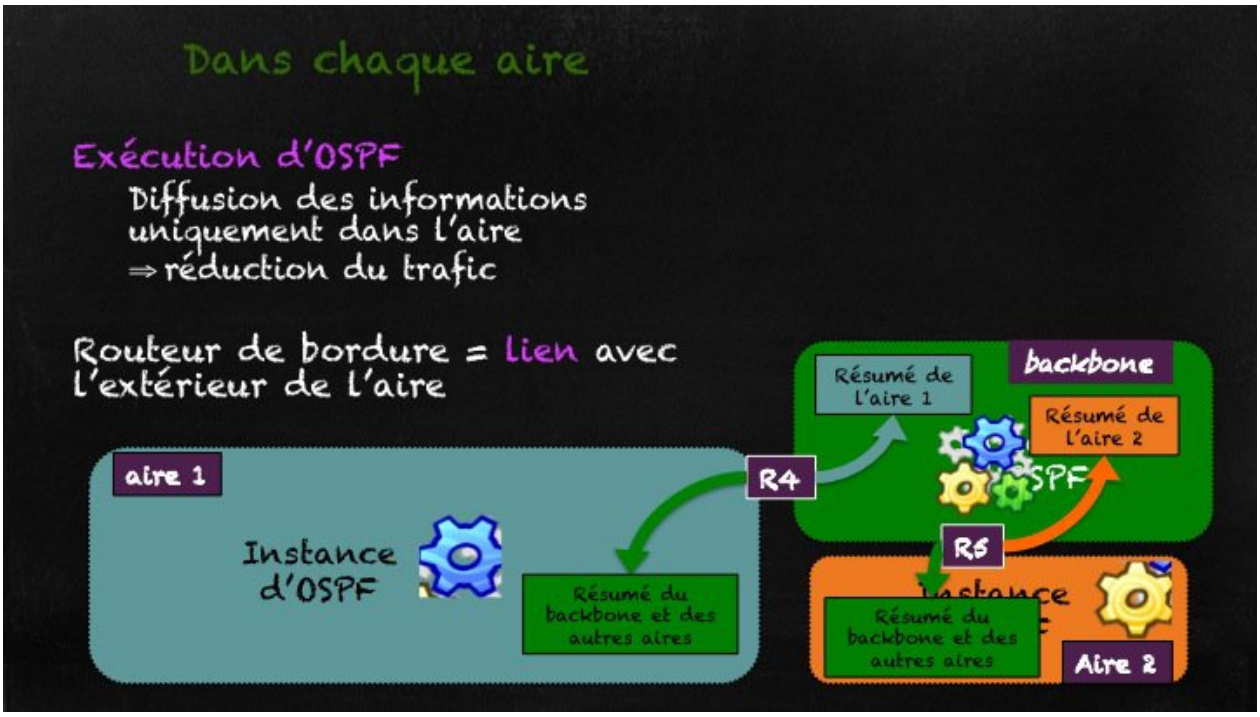
La structuration en aire



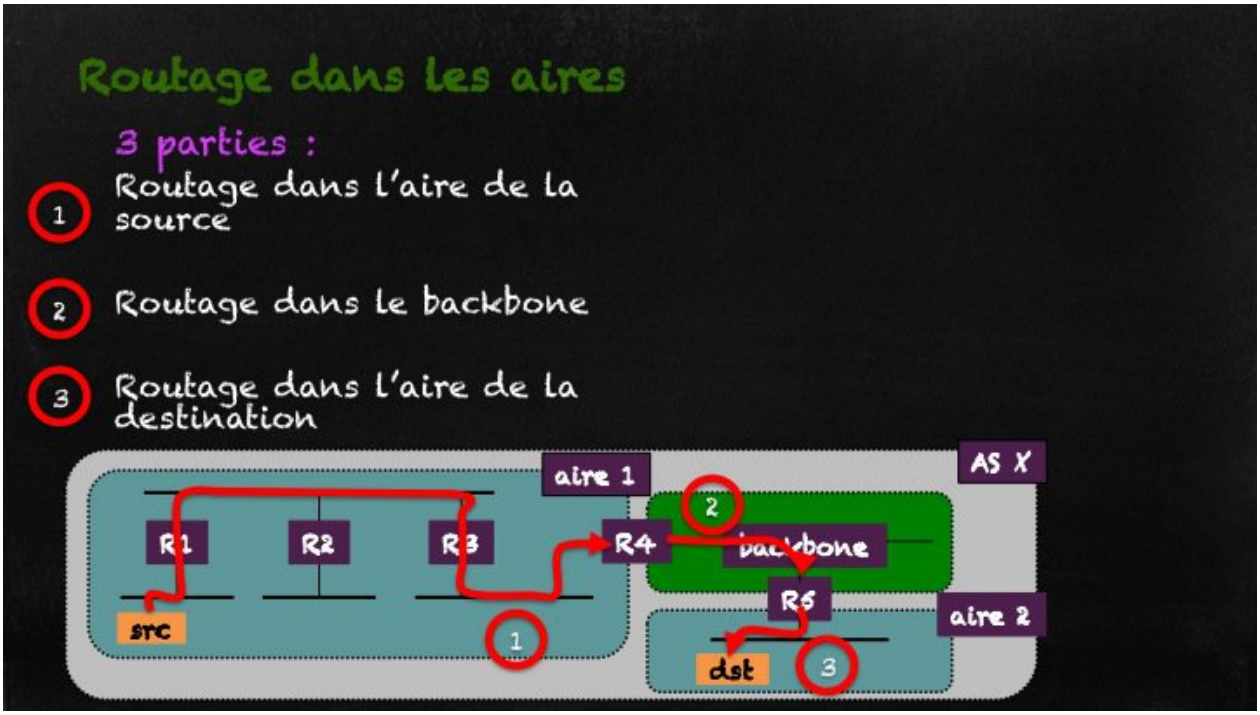
Une façon de résoudre ces compromis est d'organiser le réseau en aires : des zones exécutant chacune le protocole OSPF. Cela permet de réduire le nombre de destinataires de l'inondation et de passer à l'échelle. Ceci est particulièrement important car OSPF est destiné à de grand réseaux. OSPF, pour réduire l'impact de l'inondation en nombre de messages, incite l'administrateur à structurer le réseau en le découpant en aires contenant un sous-ensemble de réseaux, de stations, et des routeurs pour les connecter.



OSPF introduit deux niveaux de hiérarchie : les aires, et un *backbone* (appelé aire 0) qui les interconnecte. Chaque aire dispose d'au moins un routeur de bordure qui appartient à la fois à l'aire et au *backbone*. Son rôle est de faire l'interface entre les deux. La répartition des réseaux et des équipements dans les aires relève du savoir-faire de l'administrateur du réseau. Pour réduire le trafic dû au protocole, il essaiera de rassembler des réseaux ayant des plages d'adresses contiguës.



Dans chaque aire, on exécute une instance d'OSPF. Les échanges sont restreints à l'aire, et le routeur de bordure transmettra un résumé des états de liens dans le *backbone*. Cela permet de cacher certains changements de routes qui n'ont pas besoin d'être annoncés en dehors de l'aire.
Dans l'aire 0, les routeurs échangent les informations sur les réseaux de l'aire 0 et les résumés d'informations injectés par chaque routeur de bordure. Chaque routeur de bordure peut alors annoncer dans son aire les routes apprises dans le *backbone*.



La structuration en aire simplifie le routage :

- Si la destination est dans l'aire, on achemine le trafic au sein de l'aire.
- Sinon, le routage se fait en 3 parties :
  - une première partie pour acheminer le trafic vers le routeur de bordure de l'aire ;
  - une seconde partie pour acheminer le trafic dans le *backbone* jusqu'au routeur de bordure de l'aire contenant la destination ;
  - une troisième partie dans l'aire destination entre le routeur de bordure et la destination finale.

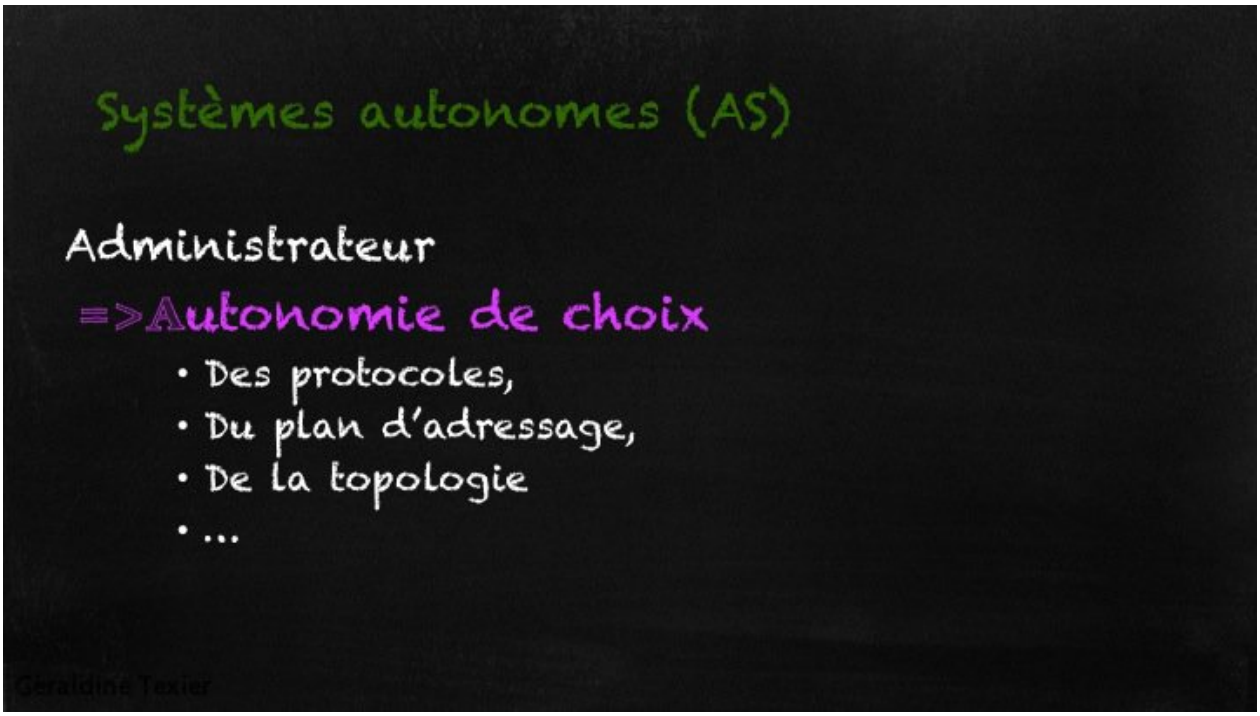
En résumé

Dans cette séance, nous avons vu un protocole de routage, OSPF, qui définit des formats de données, des échanges de messages, et utilise l'algorithme de Dijkstra pour choisir les routes. Nous avons vu que les protocoles de routage doivent faire des compromis entre la précision et le trafic qu'ils génèrent. OSPF est destiné à des réseaux de grande taille et force l'administrateur à structurer le réseau ; ce qui permet le passage à l'échelle.

Séance 5 : L'Internet et le routage entre les réseaux

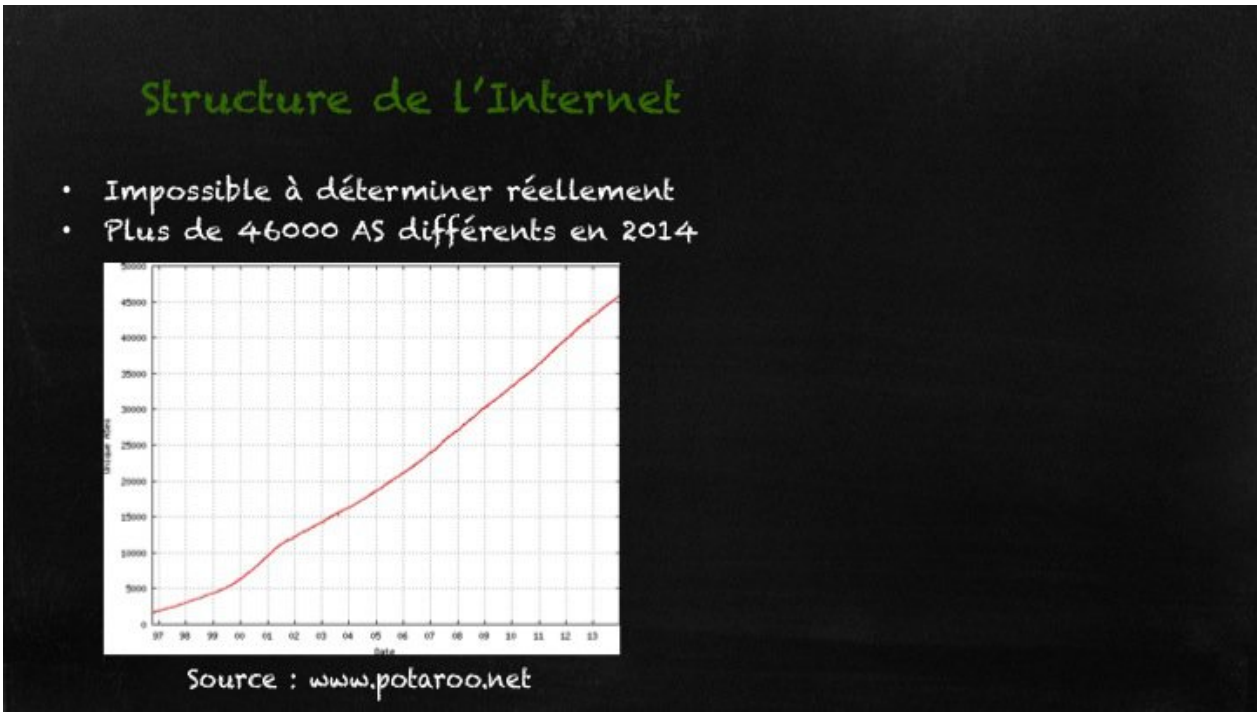
Introduction

L'Internet est formé par l'interconnexion de nombreux réseaux. Il est impossible de le gérer de façon centralisée.

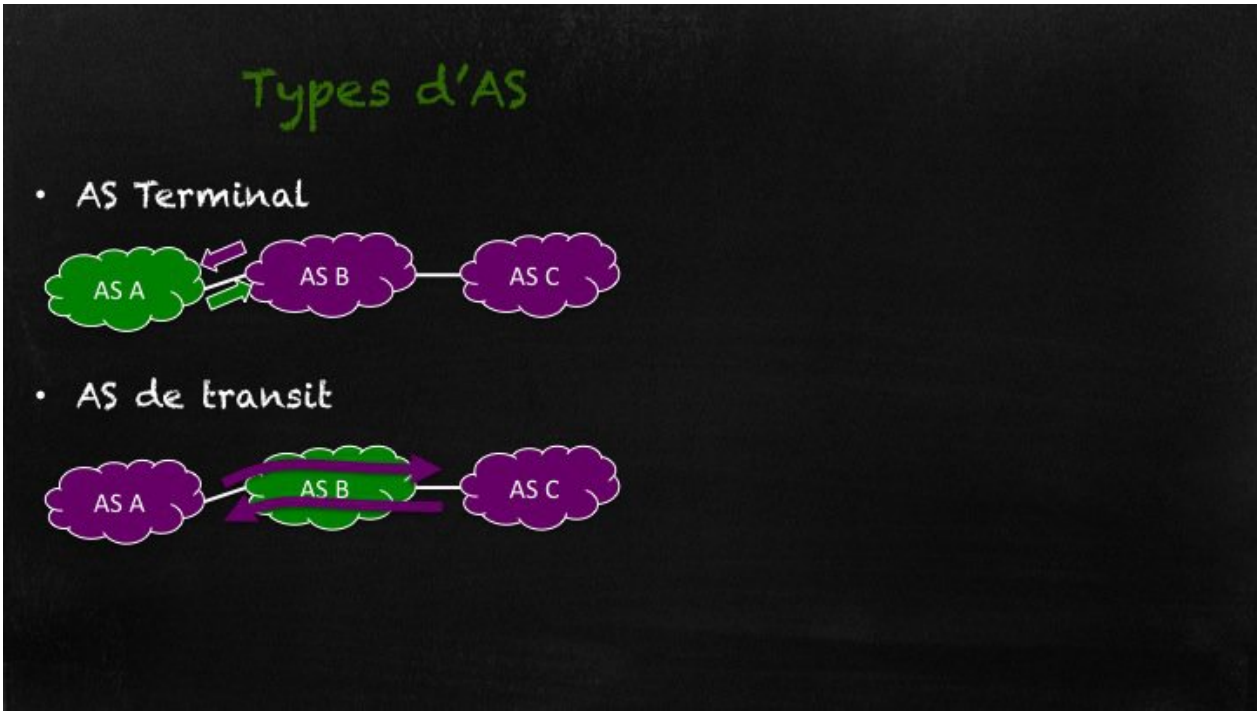


Ces réseaux forment des systèmes autonomes sous la responsabilité **d'un administrateur**. Celui-ci dispose d'une complète autonomie de gestion et définit, entre autres, son plan d'adressage et sa topologie. Il choisit le protocole de routage à utiliser à l'intérieur de son réseau, sa métrique, etc.

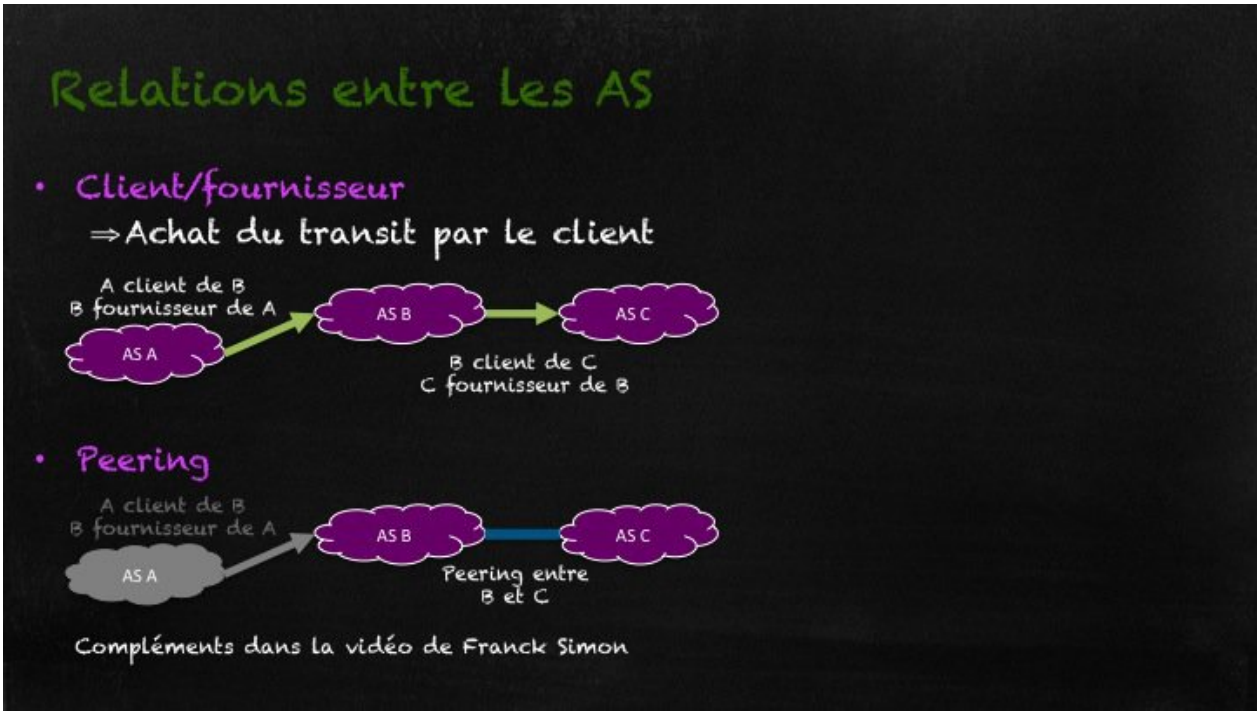
Un système autonome, aussi appelé AS, est désigné par un numéro court sur 2 octets.
La gestion des numéros et des adresses liés à l'Internet est faite par l'IANA (Internet ***Assigned Numbers Authority)***, un organisme mondial. L'IANA délègue son autorité à des organismes régionaux : des RIR (Regional Internet Registry) qui attribuent les numéros d'AS et les plages d'adresses dans leur zone géographique.



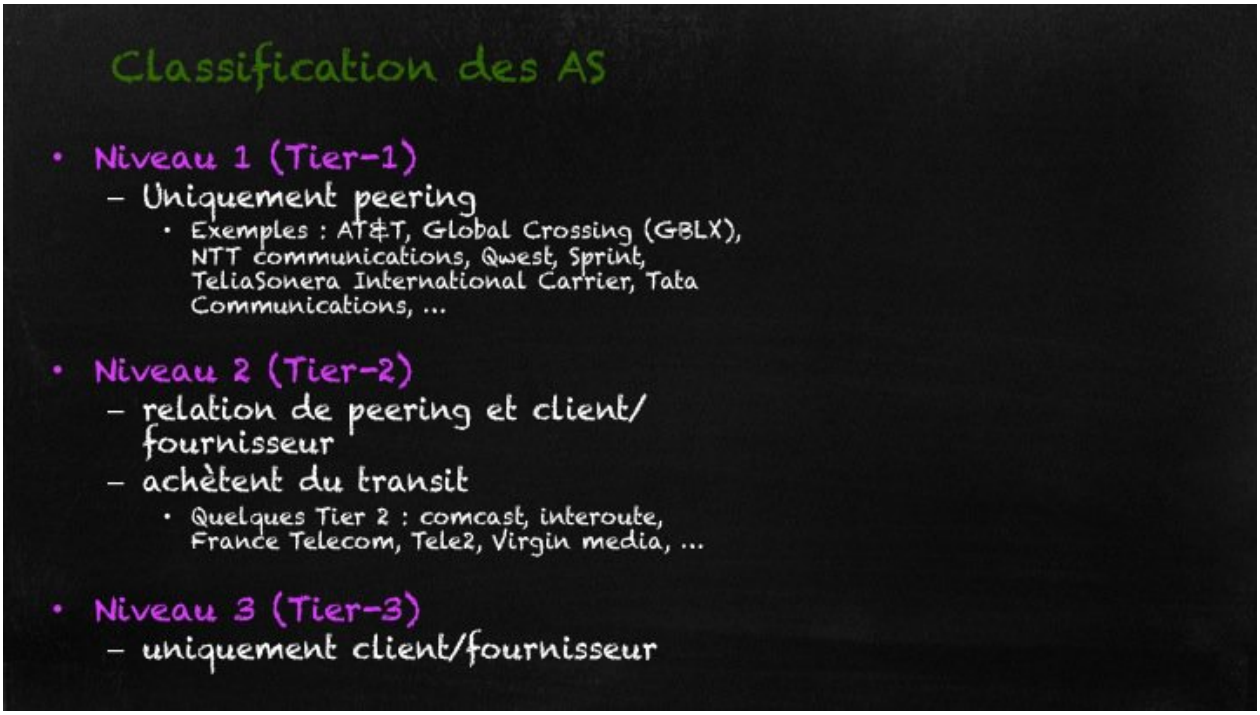
Du fait de son fort développement, il est impossible de connaître la taille et la topologie de l'Internet. Cependant, des mesures sont faites pour en avoir une vision approximative. Sur le site potaroo.net, on peut trouver une étude qui a identifié 46000 AS différents. Le graphe qui trace le nombre d'AS découverts entre 1997 et 2013 montre que ce nombre ne cesse de croître rapidement.



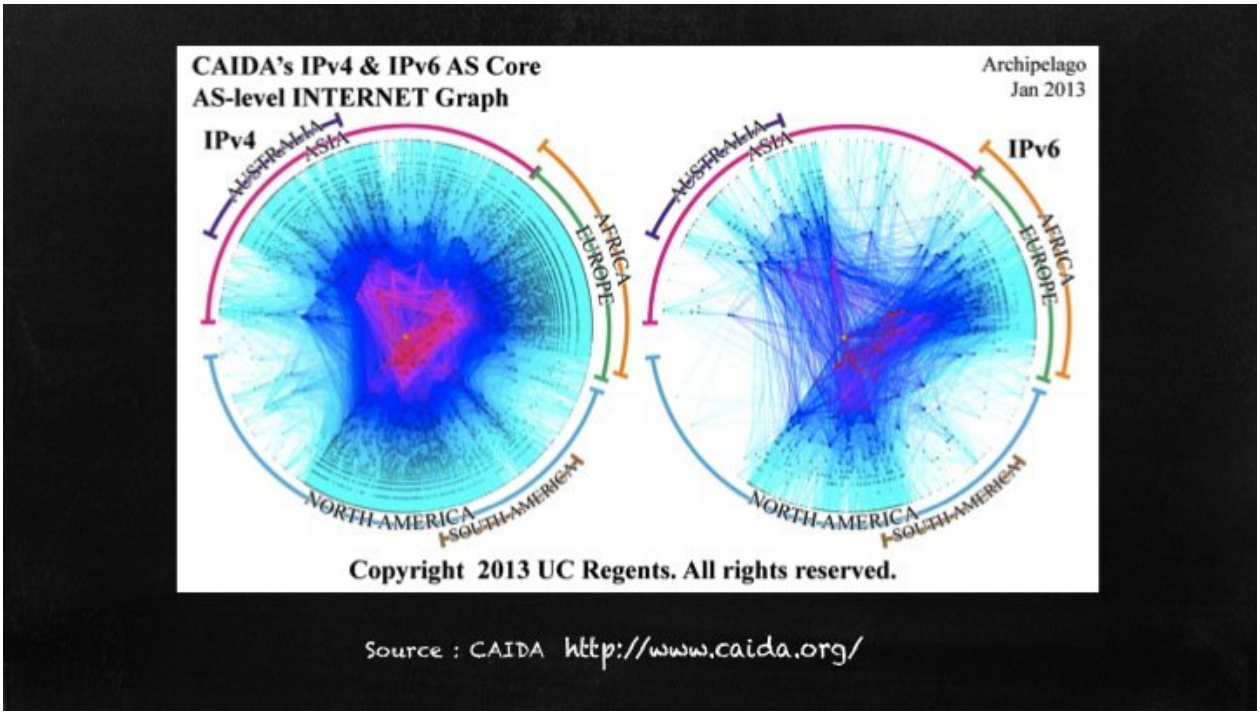
Les AS peuvent être terminaux ou de transit. S'ils sont terminaux, ils génèrent ou consomment du trafic alors que les AS de transit utilisent une partie de leur bande passante pour acheminer du trafic entre ses AS voisins.



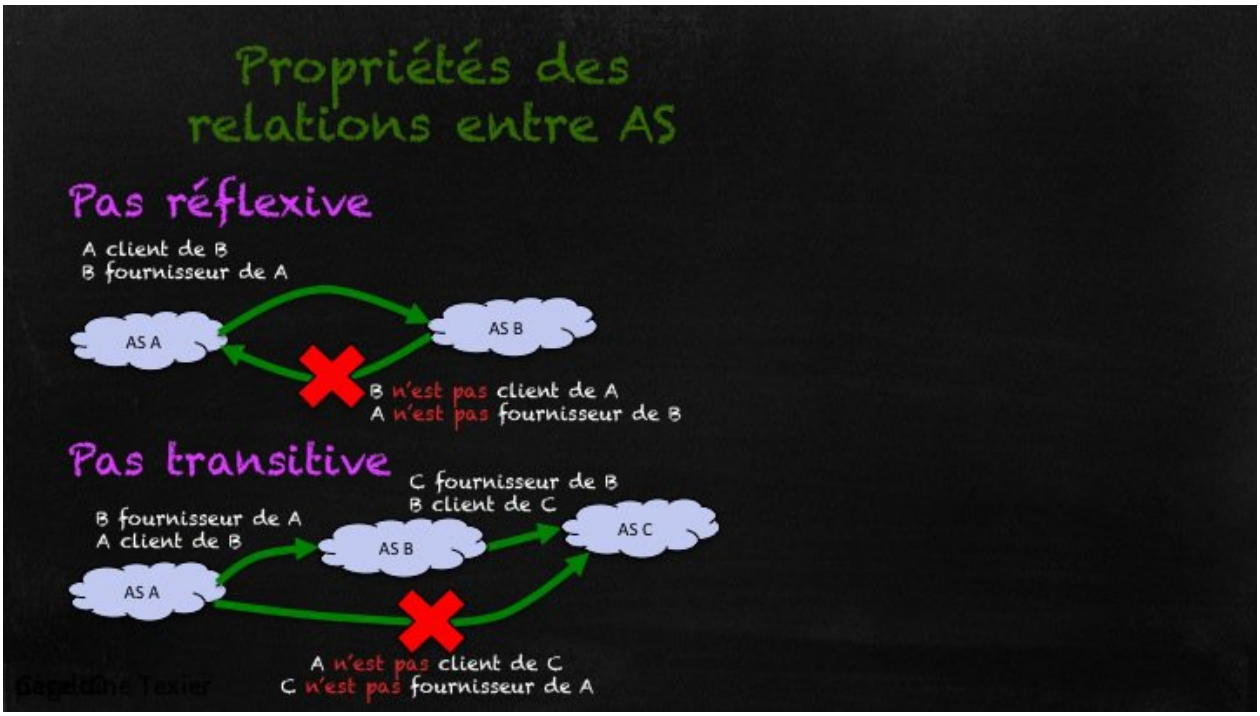
Les relations entre AS sont principalement de deux types : client/fournisseur ou *peering*. Les clients souscrivent à une offre de service auprès d'un fournisseur qui leur vend un accès à l'Internet. Quand les deux AS échangent du trafic dans des proportions comparables, il est plus pratique de négocier un accord de peering grâce auquel les deux AS échangeront librement et gratuitement du trafic. Nous reparlerons du peering lors de vidéo de Franck Simon de France IX : un point d'échange où le *peering* peut être mis en œuvre.



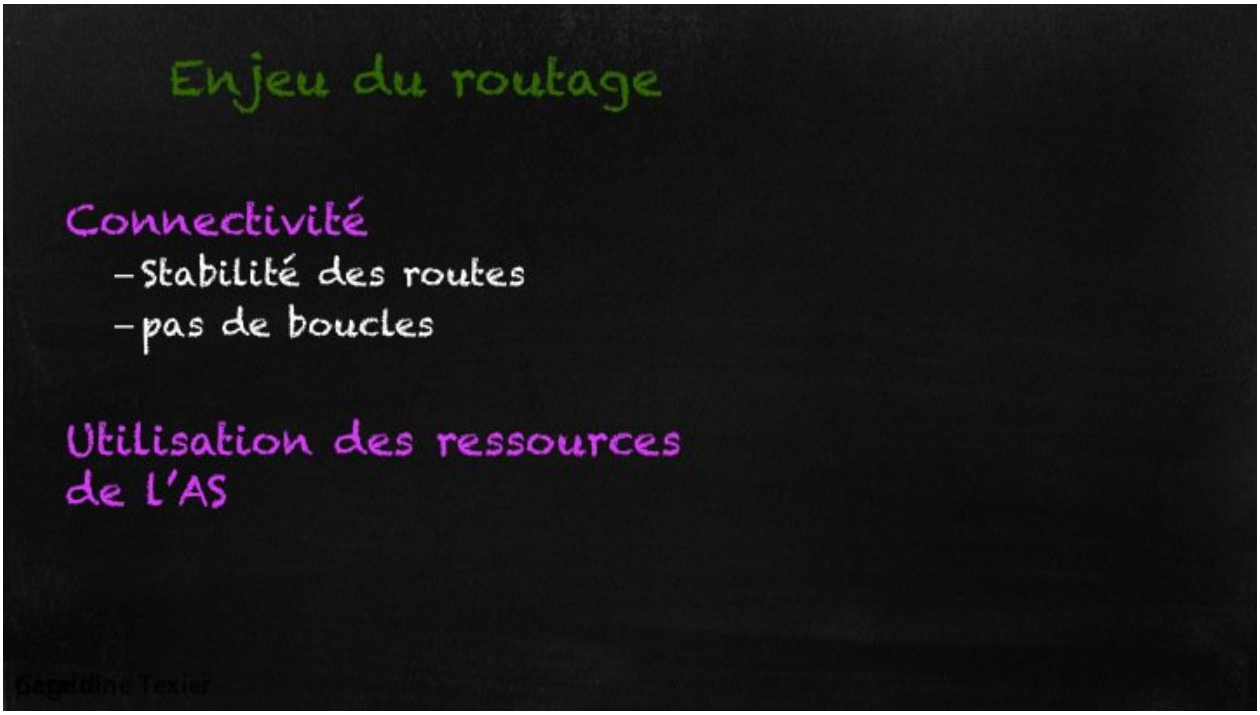
Ces relations sont à l'origine d'une classification des systèmes autonomes selon 3 niveaux. Dans le premier niveau, les AS appelés *Tier 1* sont tous complètement interconnectés par des relations de *peering*. Ces grands fournisseurs de service réseau sont par exemple AT&T, Global Crossing, Level 3, NTT, Sprint, Tata, ...
Les AS de niveau 2 (ou *Tier 2*) utilisent à la fois des relations de *peering* et des relations client/fournisseur pour acheter du transit. On peut trouver Comcast ou Tele2...
Les AS de niveau 3 (ou *Tier 3*) n'utilisent que des relations de client/fournisseur.



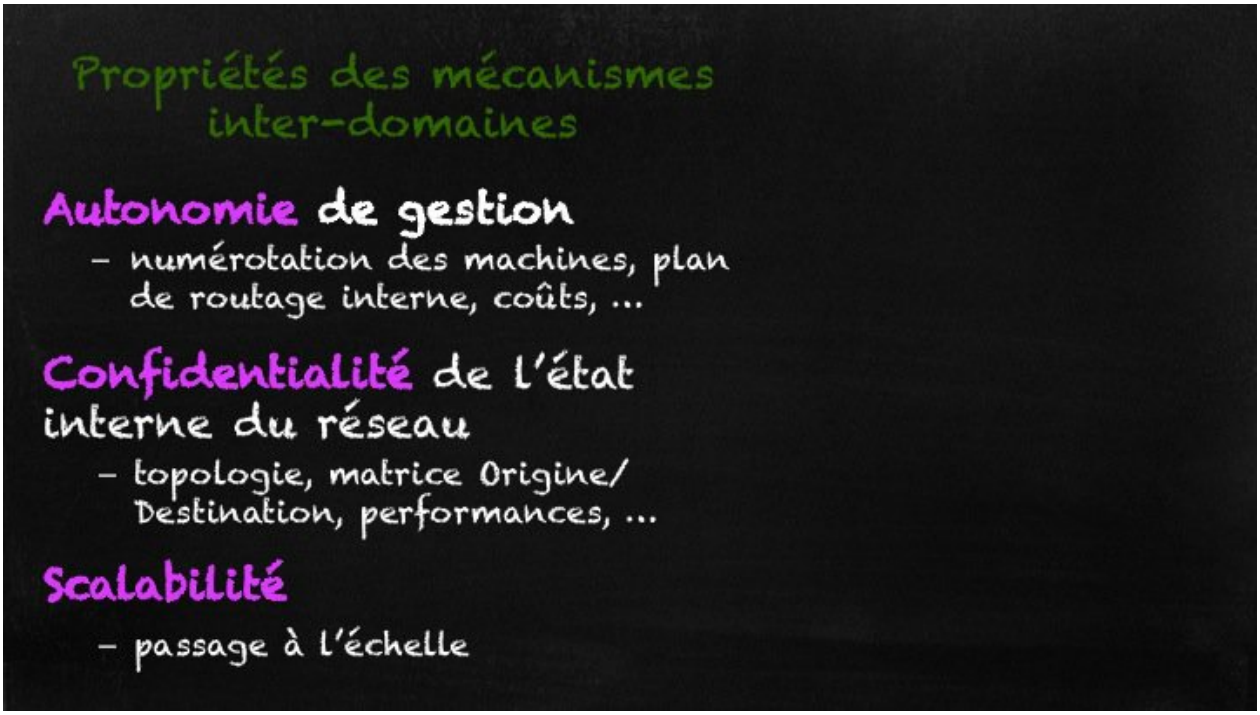
Ce graphique permet de visualiser les relations entre AS. Il est disponible sur le site de CAIDA et a été généré grâce à des mesures faites avec l'outil Archipelago. La partie rouge représente les AS ayant le plus de voisins ; c'est-à-dire le cœur de l'Internet. Si on regarde la version du graphe comportant le nom des AS, sur le site de CAIDA, on peut s'apercevoir que les Tier1 sont dans cette partie rouge.



Le routage dans l'Internet est complexe car il doit refléter les relations entre les AS et, par-dessus tout, respecter les politiques de routages choisies par l'administrateur de chaque AS. En effet, les relations client/fournisseur ne sont ni réflexives ni transitives : le fournisseur d'un AS n'est pas le fournisseur de son client.

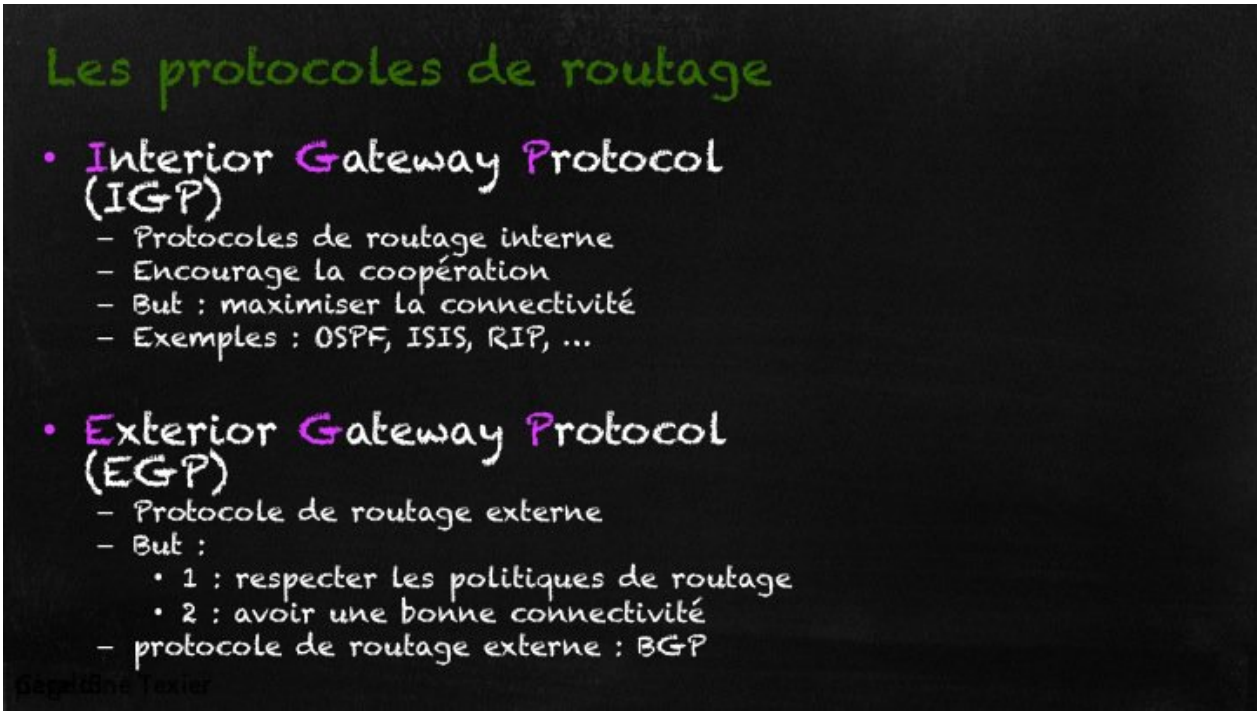


Ainsi, l'enjeu du routage entre les AS est avant tout l'utilisation des ressources de l'AS, tout en garantissant la connectivité, en favorisant la stabilité des routes, et en évitant les boucles. Accepter un trafic en transit consomme de la bande passante de l'AS.



Les mécanismes mis en œuvre entre les AS doivent vérifier trois grandes propriétés : l'autonomie, la confidentialité et la scalabilité.

- Un AS doit disposer de son autonomie de gestion. Le mécanisme ne peut imposer à l'AS une numérotation des machines, un plan d'adressage, un protocole interne, une métrique... C'est la propriété d'**autonomie**.
- La **confidentialité** indique que l'état interne de l'AS ne doit pas être diffusé. Ainsi, on ne peut diffuser la topologie de l'AS, les renseignements sur les points d'entrée ou de sortie de son trafic, ses performances.
- Enfin, tout mécanisme entre les AS doit **passer à l'échelle** et limiter l'impact sur ses performances du grand nombre d'AS dans l'Internet.

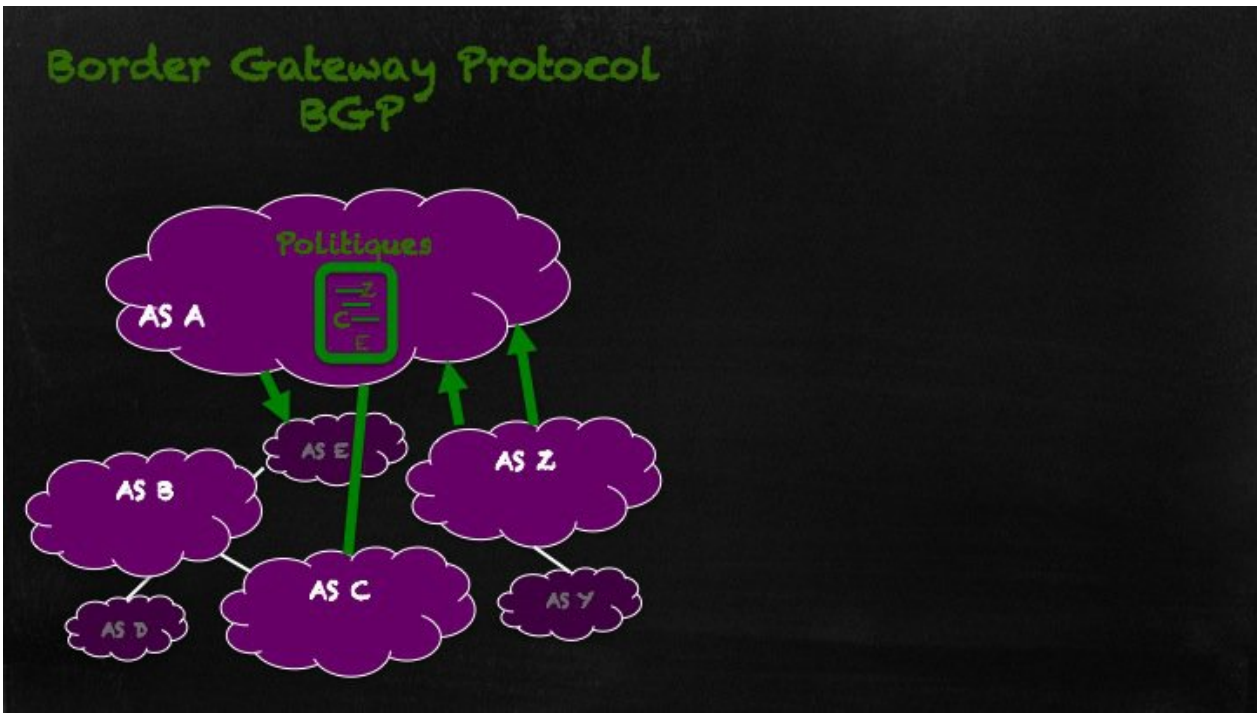


Les protocoles de routage sont divisés en deux familles : les IGP *Interior Gateway Protocol* (tels qu'OSPF, IS IS) qui régissent le routage au sein d'un AS, en intradomaine, et l'EGP *Exterior Gateway Protocol* (BGP) qui définit le routage entre les AS, en interdomaine.

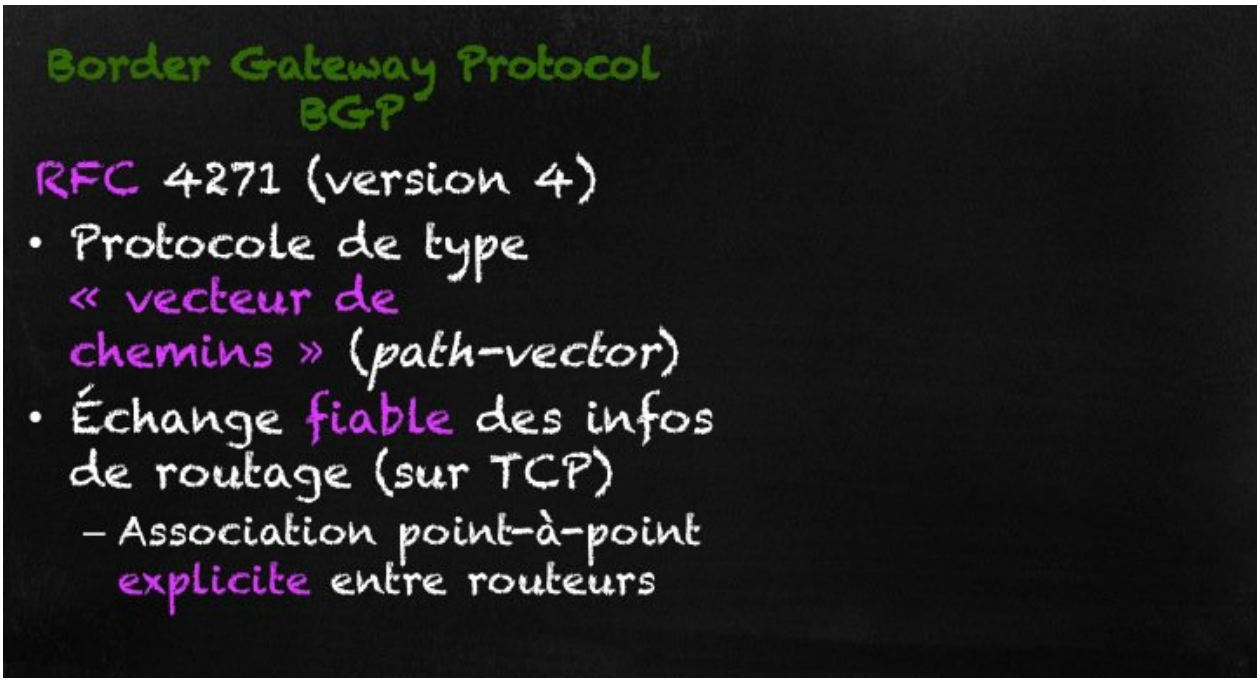
L'IGP encourage la coopération entre les routeurs de l'AS pour avoir la meilleure connectivité, contrairement à l'EGP qui respecte avant tout les politiques de routage.
Il vaut mieux ne pas offrir de connectivité à un AS voisin si aucune relation de client/fournisseur ou *peering* n'a été établie.
Afin de pouvoir mettre en œuvre le routage entre les AS, tous doivent exécuter le même protocole : BGP *Border Gateway Protocol*.

Séance 7 : BGP

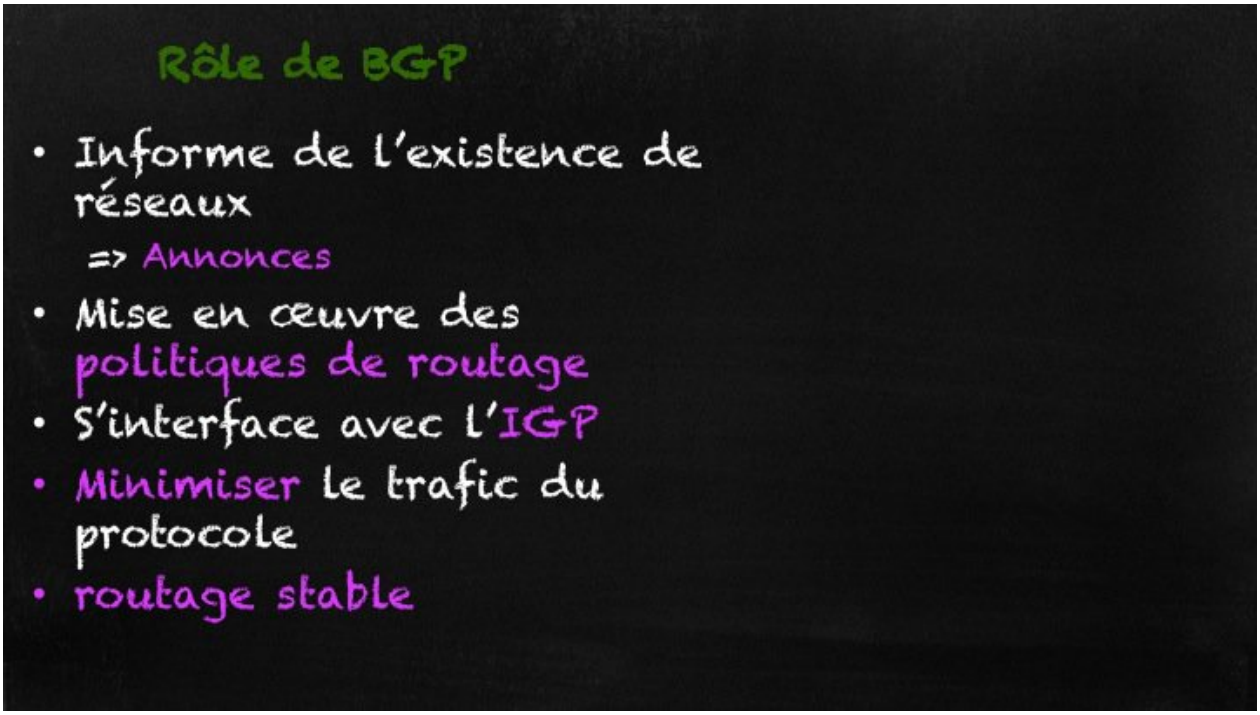
Introduction



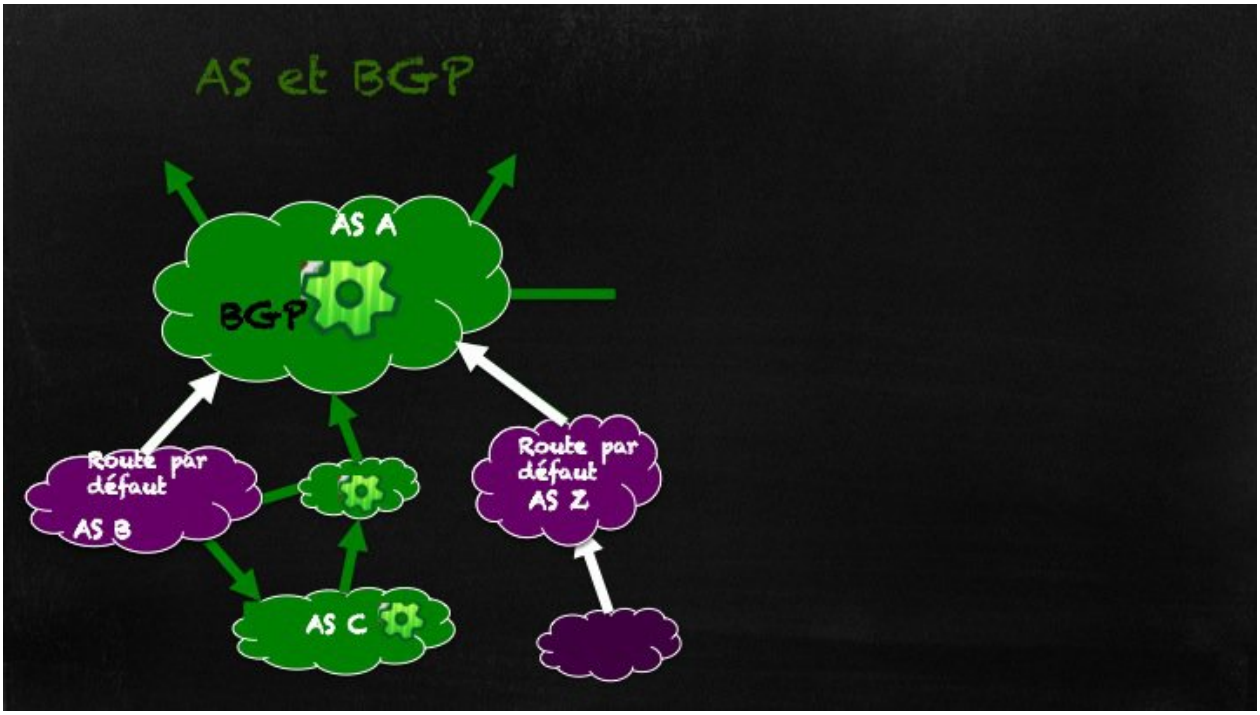
BGP (Border Gateway Protocol) régit le routage entre les AS en cohérence avec les politiques de routage définies par les administrateurs.



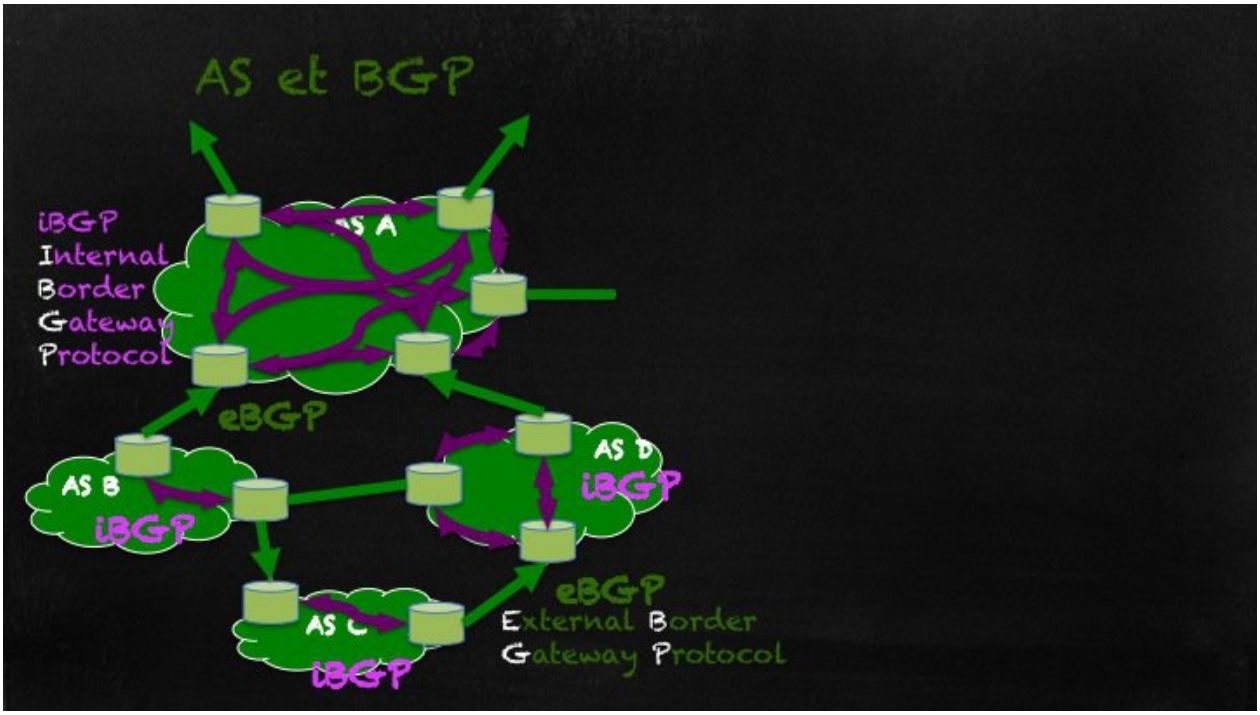
Il est standardisé à l'IETF et met en œuvre un protocole de type "vecteur de chemin" (*Path Vector*). Il nécessite un établissement explicite d'une session d'échanges fiables entre deux routeurs grâce à TCP.



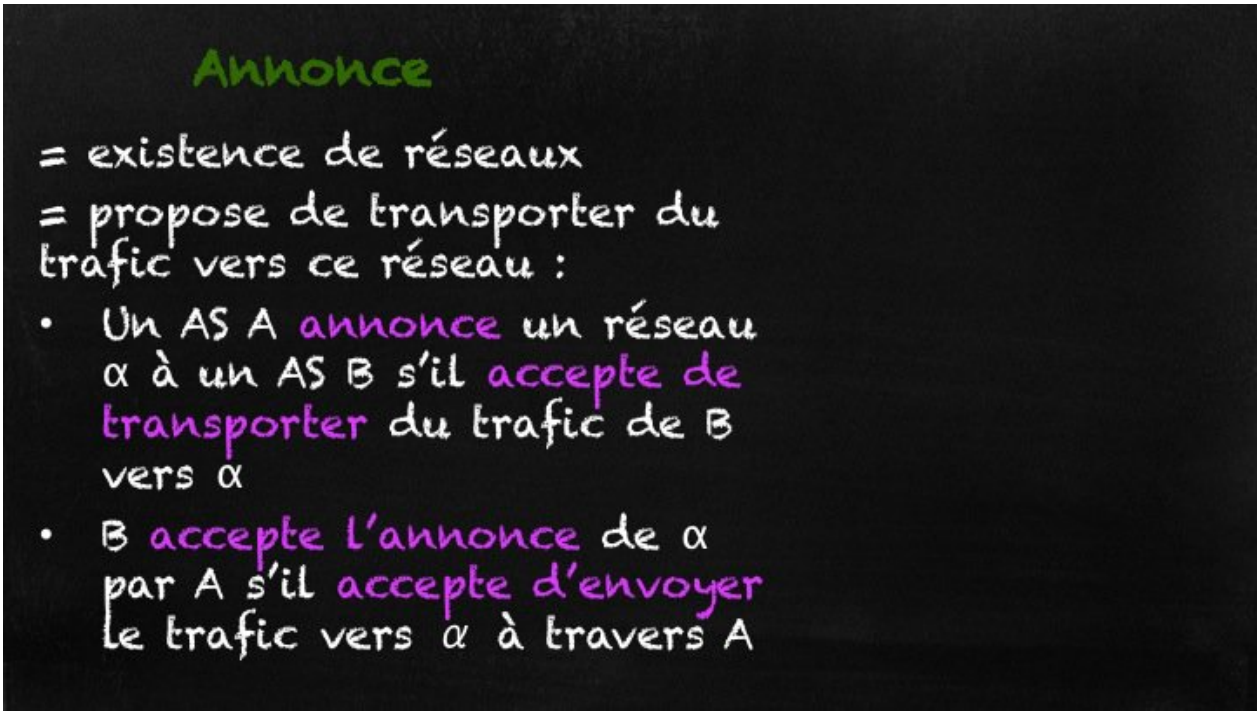
BGP a pour rôle d'informer de l'existence de réseaux en transmettant des annonces, de mettre en œuvre la politique de routage de l'AS, de s'interfacer avec le protocole de routage interne de l'AS tout en minimisant le trafic généré sur les liens par le protocole et en favorisant un routage stable.



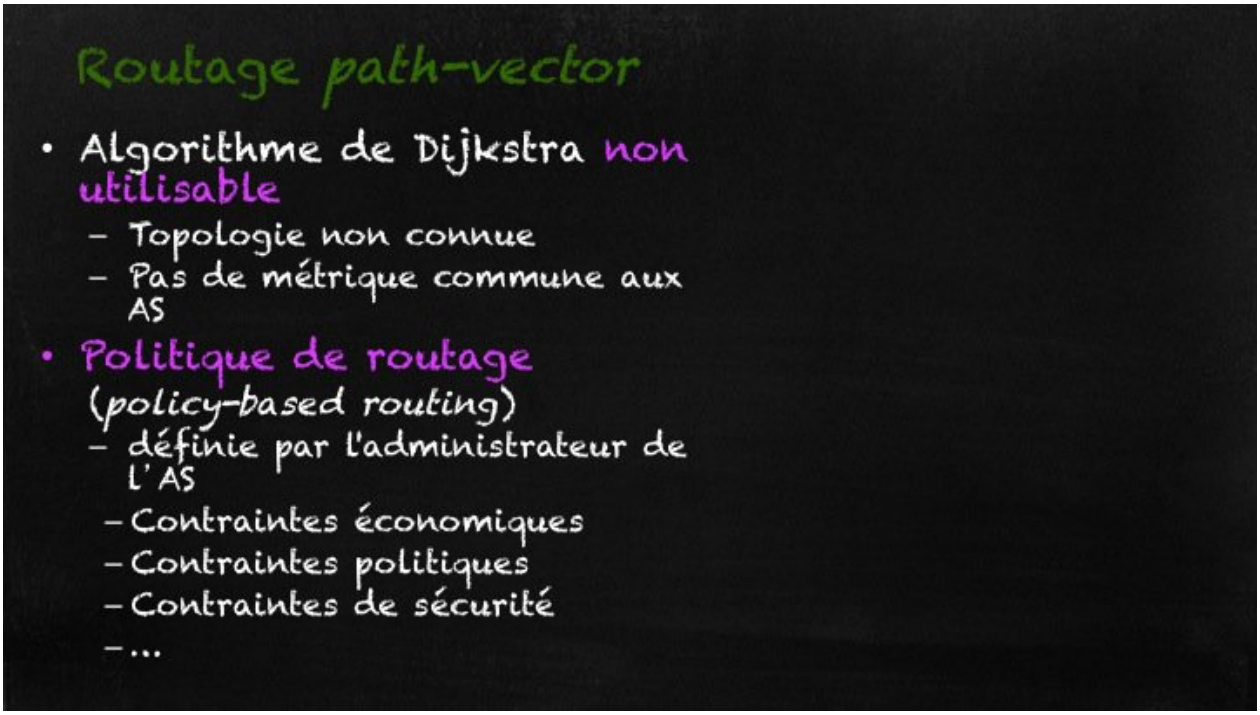
Un AS peut être connecté à un seul fournisseur de service réseau. Il n'a besoin que d'une route par défaut vers ce voisin pour atteindre l'Internet. **Le recours à BGP devient nécessaire quand un AS met en œuvre des connections avec plusieurs fournisseurs de service réseau (on dit qu'ils sont multidomiciliés) ou lorsqu'ils offrent un service de transit**. Il faut alors choisir les routes qui seront injectées dans la table de routage.



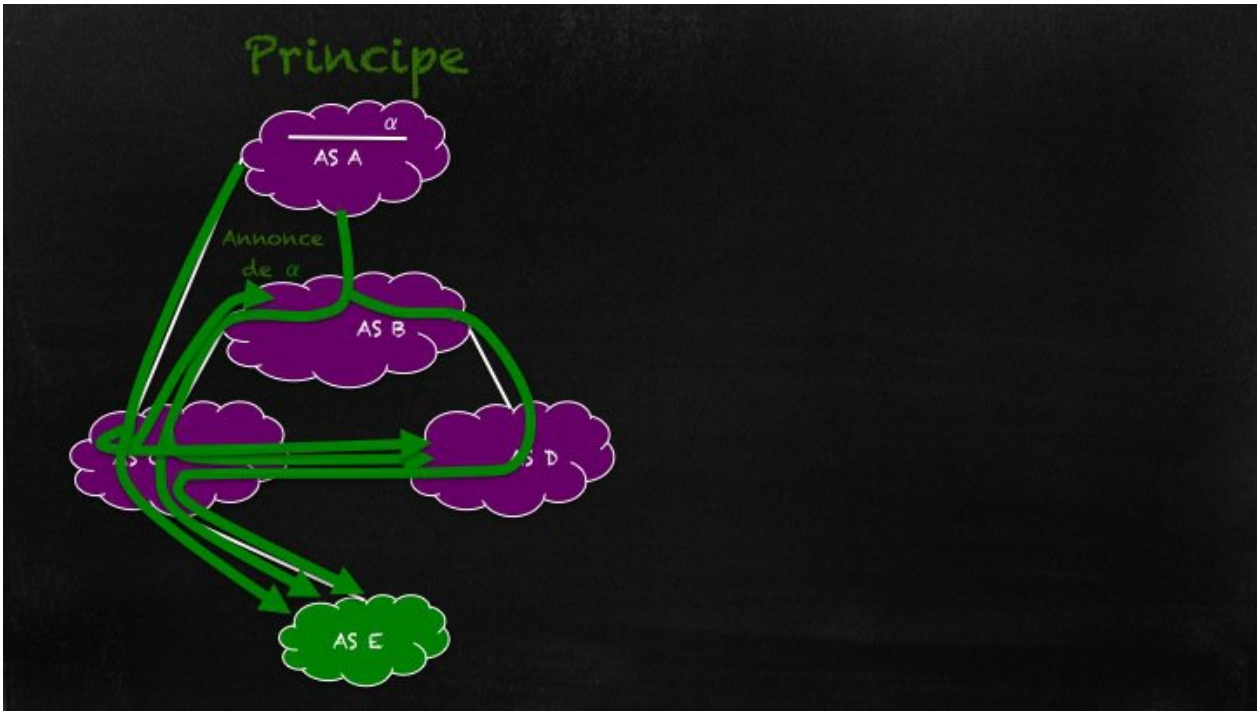
Le protocole BGP établit explicitement des relations externes avec certains routeurs des AS voisins et des relations internes entre tous les routeurs exécutant BGP. Ces routeurs de bordure d'AS dialoguent et échanges les annonces apprises par le protocole de routage interne ou des AS voisins.



Transmettre une annonce n'est pas anodin. Cela signifie que l'on accepte de transporter du trafic à destination du réseau annoncé. Ainsi, quand un AS A annonce un réseau alpha à un AS B, il indique implicitement qu'il accepte de transporter le trafic envoyé par B à destination de Alpha.
Quand un AS B reçoit l'annonce du réseau Alpha en provenance de l'AS A, il choisit de l'intégrer dans son routage ou non. Il peut refuser, si sa politique ne permet pas de passer par l'AS A pour joindre le réseau Alpha, ou s'il a déjà une route meilleure vers ce réseau.
BGP met en œuvre un routage par omission. En effet, un AS peut avoir une connaissance totale des préfixes de l'Internet s'il est bien connecté. Par contre, il n'annonce pas tout. L'enjeu d'un bon routage est la connectivité ; mais surtout, l'emploi des ressources de l'AS par ses voisins si le routage est mal fait.



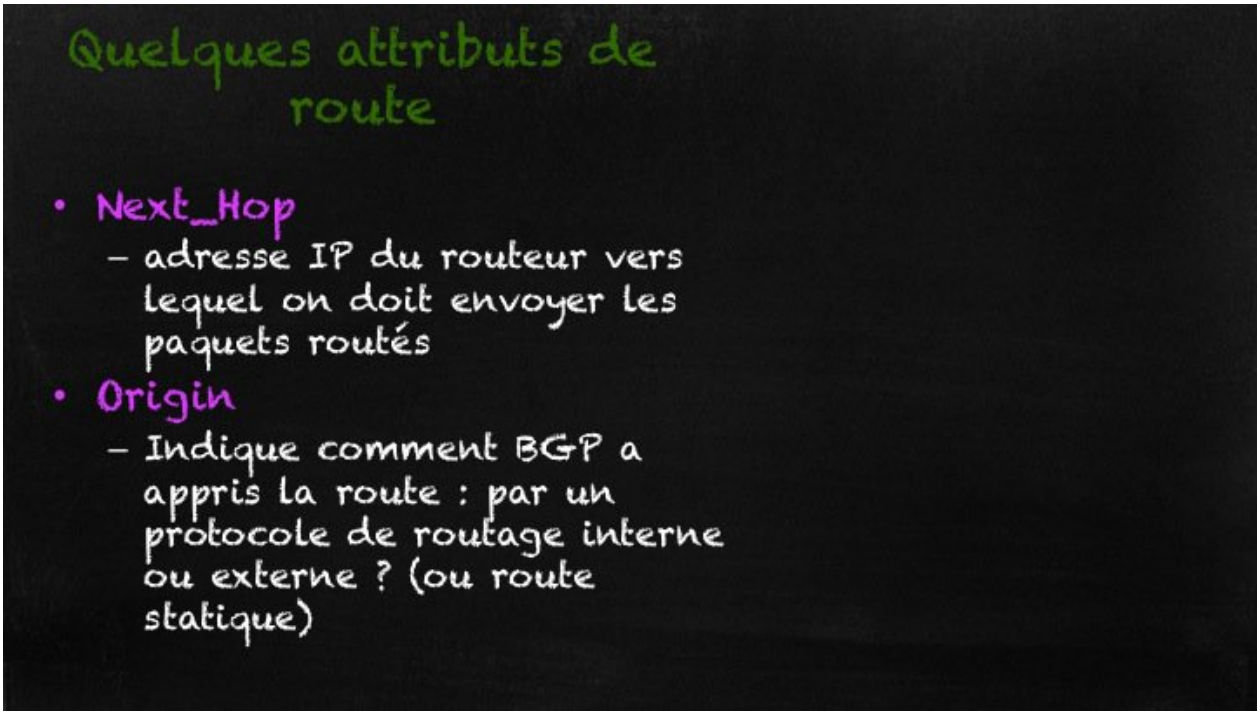
Pour choisir un chemin, on ne peut se contenter d'appliquer une métrique et d'effectuer un algorithme de Dijkstra. D'une part, parce qu'on ne peut diffuser la topologie complète de l'Internet et des AS pour déterminer le plus court chemin, d'autre part parce qu'on ne peut imposer une métrique unique à tous les AS. Enfin, les routes choisies doivent respecter les politiques de routages définies par l'administrateur. Elles peuvent refléter des contraintes économiques, politiques, de sécurité...



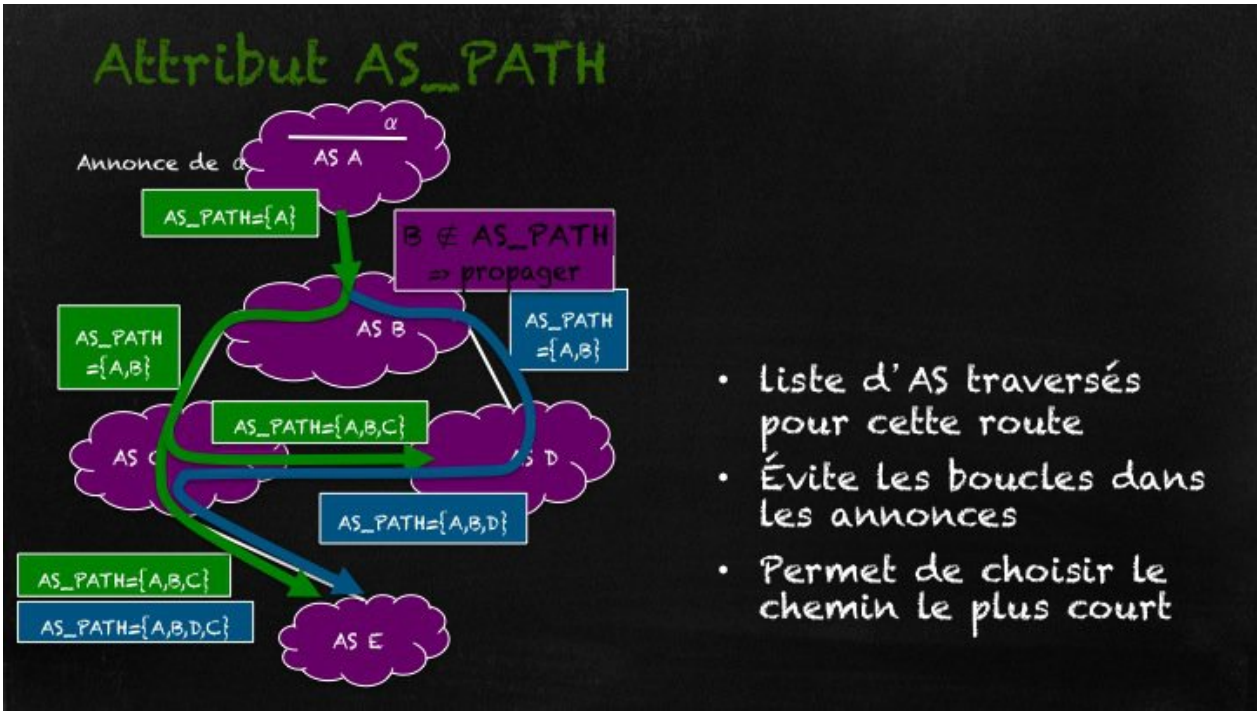
Un AS peut recevoir plusieurs annonces pour un même réseau. Pour limiter la taille des tables de routage, il n'en gardera qu'une. Dans la plupart des cas, le chemin choisi par BGP sera le plus court en nombre d'AS traversés, ou celui qui est désigné comme le meilleur après avoir appliqué une série de critères.

Pour chaque annonce, on dispose d'un ensemble d'attributs qui permettent de qualifier cette route. Certains sont obligatoires et chaque routeur doit les mettre en œuvre. D'autres sont

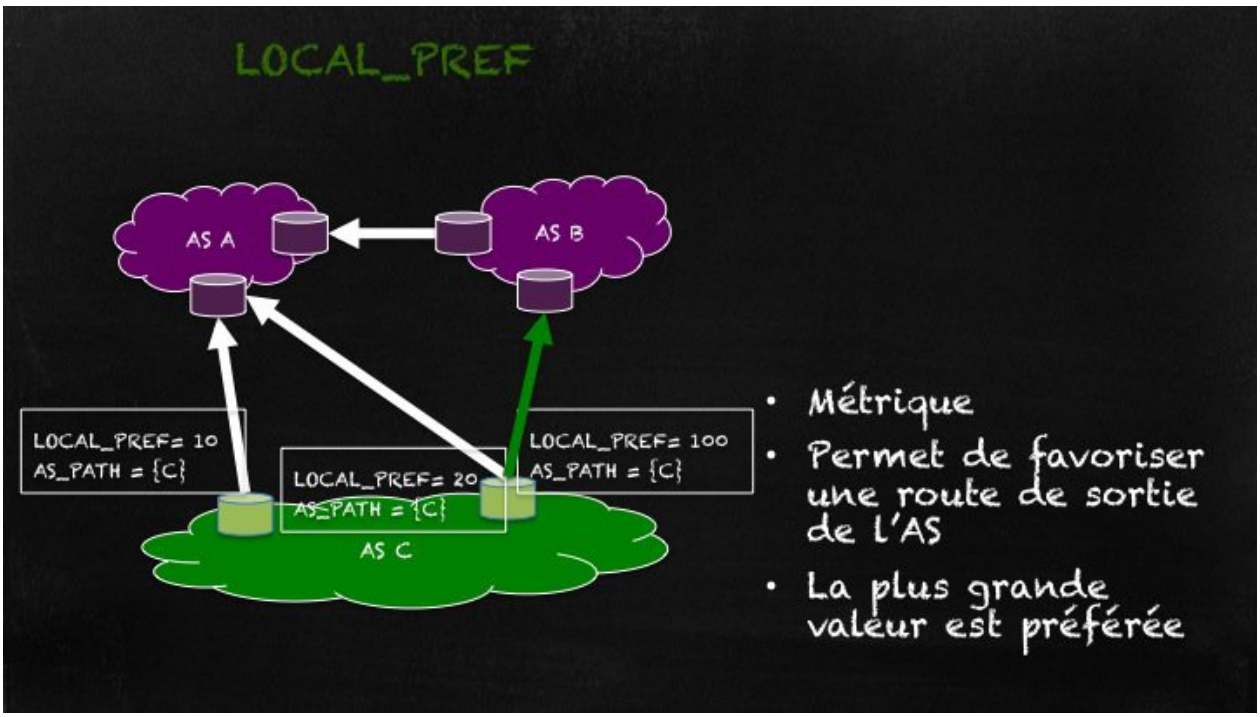
optionnels. Une partie des attributs donne une information sur la route ; d'autres indiquent la préférence d'un administrateur de réseau ; d'autres permettent d'attribuer des traitements communs à plusieurs annonces...



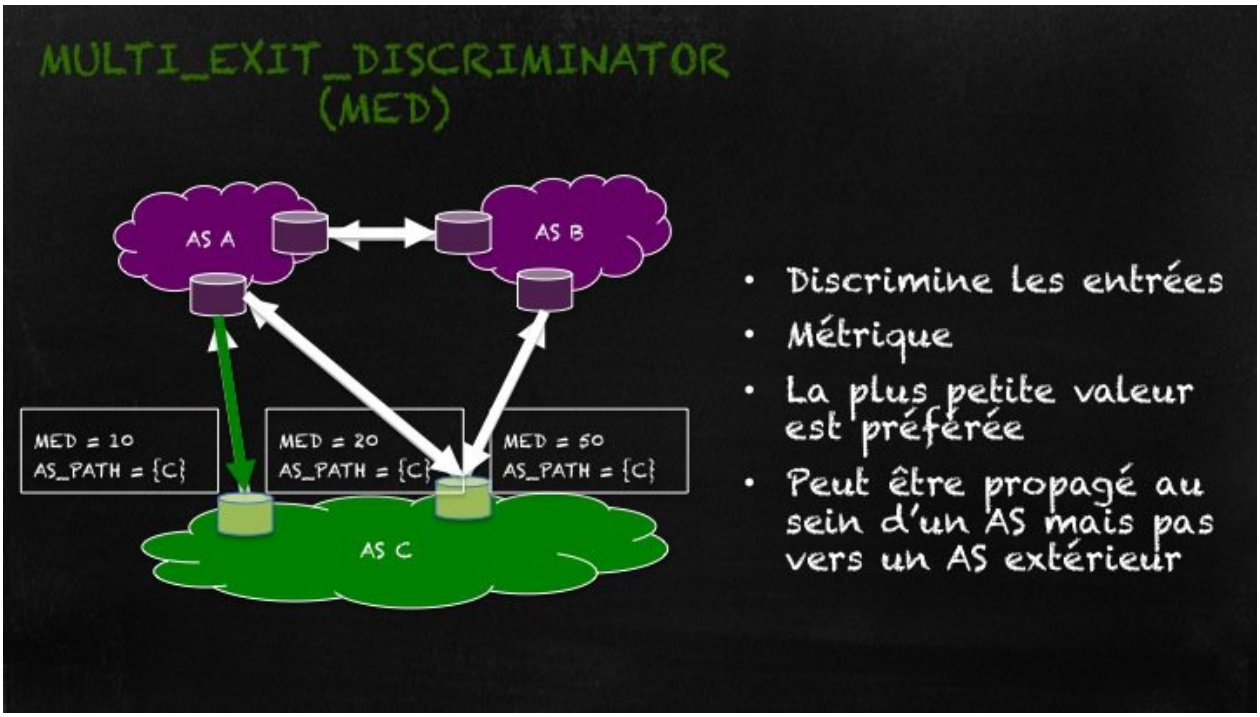
L'attribut NextHop indique l'adresse IP de l'équipement à utiliser comme prochain saut pour joindre le réseau annoncé. L'attribut Origin indique si la route a été apprise par BGP, par le protocole de routage interne, ou par un autre moyen. Cet attribut est important car apprendre la route par le protocole de routage interne garantit que le chemin en interne pour utiliser cette route est connu.



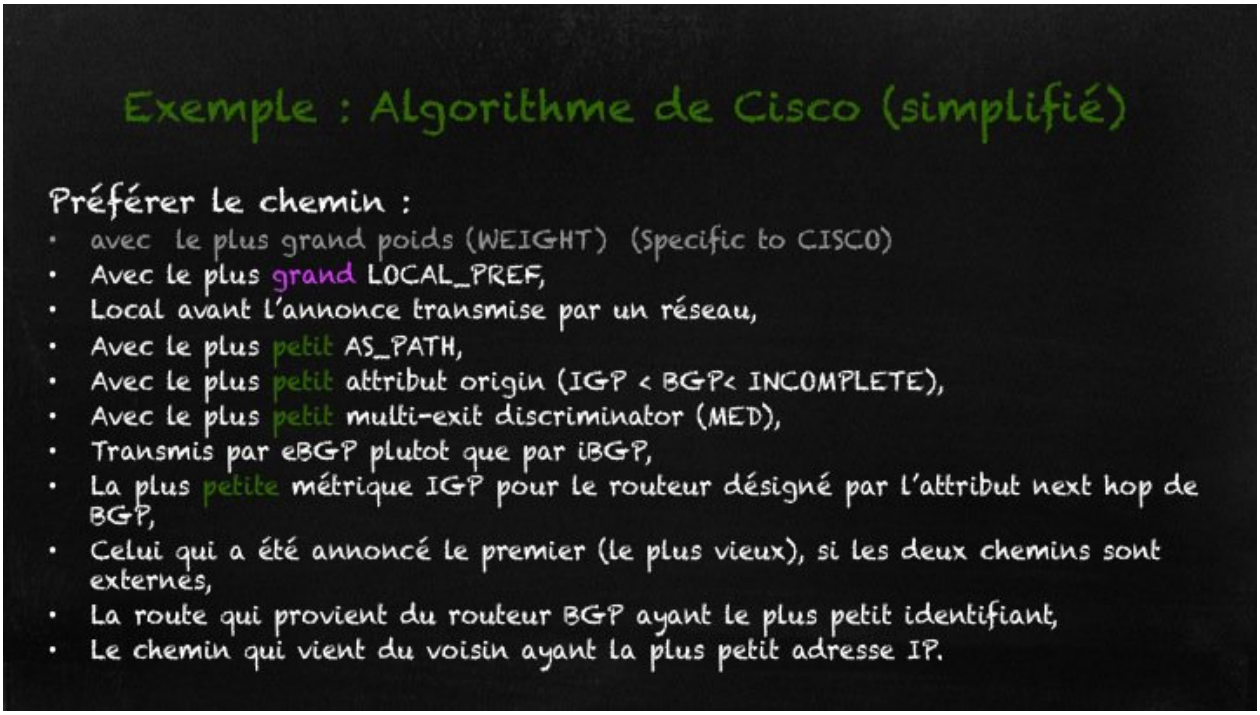
L'attribut ASPath contient la liste des systèmes autonomes parcourus par l'annonce de la route. Cela permet de choisir, parmi plusieurs chemins, la route qui traversera le moins d'AS. Cet attribut permet aussi de s'assurer que la route ne présente pas de boucle. En effet, un AS recevant cette annonce vérifie qu'il n'est pas déjà dans l'AS Path. S'il y figure déjà, il ne propagera pas l'annonce pour ne pas générer de boucle. Sinon, il pourra propager l'annonce à ses voisins si sa politique de routage le lui permet.



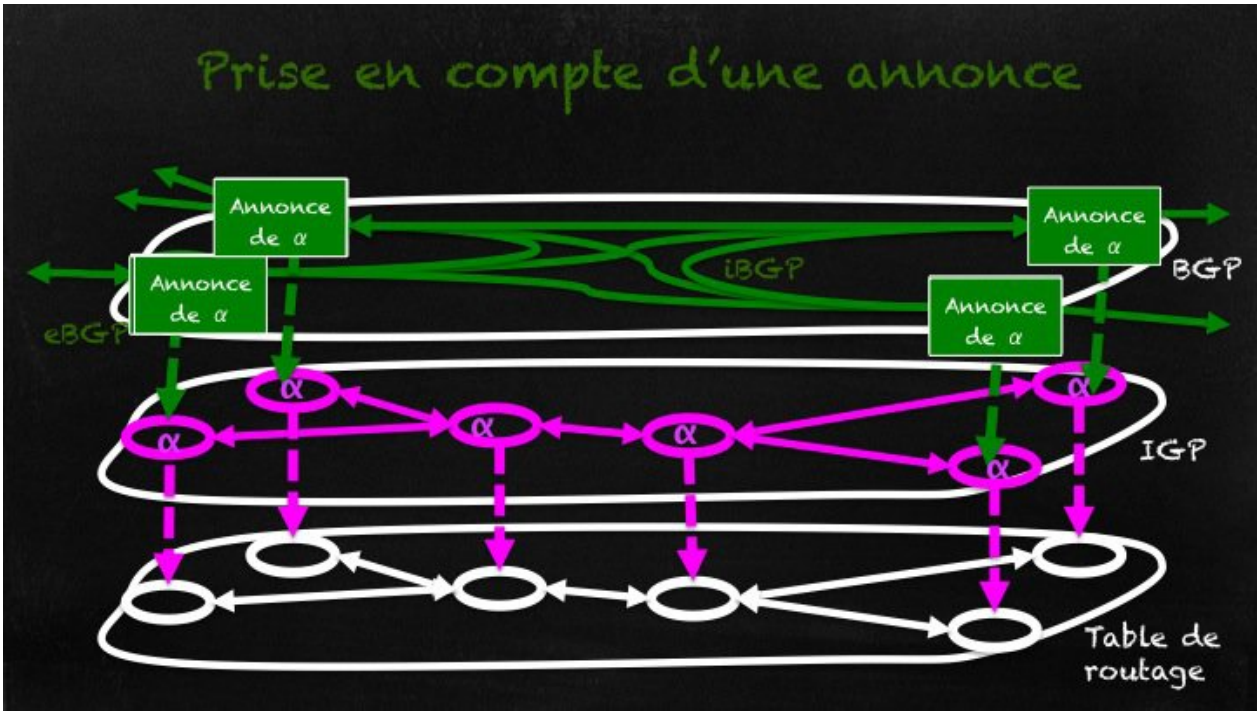
Dans un AS, lorsque plusieurs routeurs peuvent faire sortir le trafic, l'attribut LOCAL PREF permet à l'administrateur d'indiquer sa route de sortie préférée.
Chaque AS choisit ses routes de façon indépendante ; l'administrateur d'un AS ne peut imposer ses choix aux autres AS.



Lorsque plusieurs routes permettent de faire entrer du trafic dans un AS, BGP offre cependant un moyen de recommander une route plutôt qu'une autre à un AS voisin grâce à l'attribut MED.



Voici un exemple d'algorithme de sélection de chemin en BGP : Si le constructeur met en œuvre un poids pour chaque annonce, on peut préférer celle de plus haut poids. Sinon, on tiendra compte du Local Pref pour favoriser la route sortant par un routeur de bordure préféré par l'administrateur. Sinon, on gardera l'annonce ayant le plus petit ASPath. Sinon, on retiendra l'annonce faite par le protocole de routage interne plutôt que par BGP ou par une origine inconnue. Sinon, on retiendra l'annonce ayant le plus petit attribut MED, favorisant ainsi la route préférée de l'AS voisin. Si cela n'a pas suffit pour départager les annonces, on continuera à parcourir les attributs jusqu'à départager les routes.



Une fois qu'une nouvelle annonce a été retenue, le routeur de bordure du système autonome va l'annoncer :

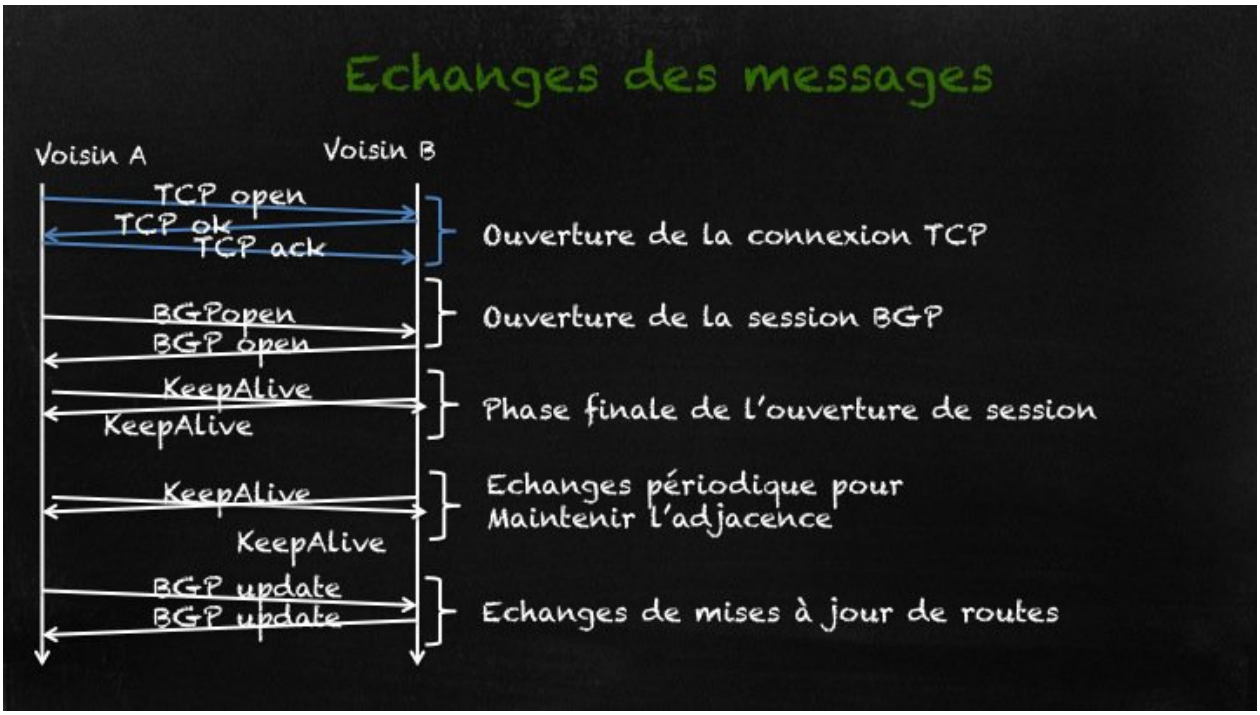
- à tous les routeurs de bordure qui participent à BGP grâce à leur liaison BGP interne ;
- éventuellement aux AS voisins grâce à ses liaisons BGP externes. Il va également injecter cette route dans le protocole de routage interne afin de l'intégrer à la table de routage.

BGP : type de messages

Message	Type	Raison
Open (ouverture)	1	Se connecter à un voisin (acquisition)
Keepalive (sonde)	4	• Acquitter un message Open • Garder le contacte avec le voisin
Update (mise à jour)	2	• Envoyer des infos sur une route • Lister des routes à éliminer
Notification	3	Notifier de la détection d'une erreur (et fermer la connexion)

BGP met en œuvre plusieurs types de messages.

- Le message d'**ouverture de connexion** BGP. Rappelez-vous que les routeurs qui échangent des informations par BGP sont explicitement paramétrés par l'administrateur de l'AS. L'ouverture de la connexion fait suite à une décision administrative d'établir une collaboration entre les deux AS et une désignation des machines pour la mettre en œuvre. Cette ouverture de connexion est acquittée par la réception d'un message KeepAlive.
- Le message **KeepAlive** est également envoyé périodiquement pour garder le contact avec le nœud BGP de l'AS voisin.
- Le message **Update** permet d'envoyer des informations sur une route et d'indiquer une liste de routes à effacer.
- Un message de **notification** peut être utilisé en cas de d'erreur avant de fermer la connexion.



Résumons les phases du protocole.
Lors de la phase d'acquisition des voisins :Un routeur veut échanger des informations.
Il envoie un message Open.Si l'autre routeur accepte la requête : envoi d'un message Keepalive.
Pendant la durée de la connexion, l'AS doit garder le contact avec ses voisins (neighbor reachability). Pour cela, il envoie un message Keepalive de façon périodique.
Dans la phase de mise à jour de la connectivité avec les autres réseaux (network reachability), chaque routeur a une base de données de réseaux qu'il peut atteindre et une route préférée vers chaque réseau. Si la base de données change, on envoie le changement à l'aide d'un message Update.

En résumé

Nous avons abordé le protocole BGP. Contrairement au protocole de routage interne, le but n'est pas d'échanger le plus d'informations possible pour améliorer la connectivité. Il faut filtrer les informations à diffuser et mettre en œuvre explicitement des relations de voisinage qui reflètent les politiques de routage et donc les accords entre les AS.

BGP a pour rôle de propager des annonces de réseaux tout en limitant la taille des tables de routage. Pour cela, on ne garde qu'une route vers une destination donnée. Le choix de la route à garder est fait grâce à l'algorithme de sélection de BGP. Nous avons vu les grands principes de BGP. Pour aller plus loin dans l'étude de ce protocole, je vous invite à consulter le références proposées sur le site du cours et les standards de l'IETF.