

Algorithmes de routage

Introduction

Définition

L'objectif du routage est de déterminer une route (i.e. un ensemble de liens à parcourir), respectant certaines contraintes, pour établir une connexion d'un noeud source vers un noeud destinataire. Le but d'un algorithme de routage est de permettre le calcul de route entre ces deux noeuds au sens d'un certain critère, et la diffusion des informations nécessaires à ce calcul.

De l'importance du routage

L'émergence des réseaux haut débit rend de plus en plus crucial le calcul de route entre source et destinataire lors de l'établissement d'une connexion. En effet:

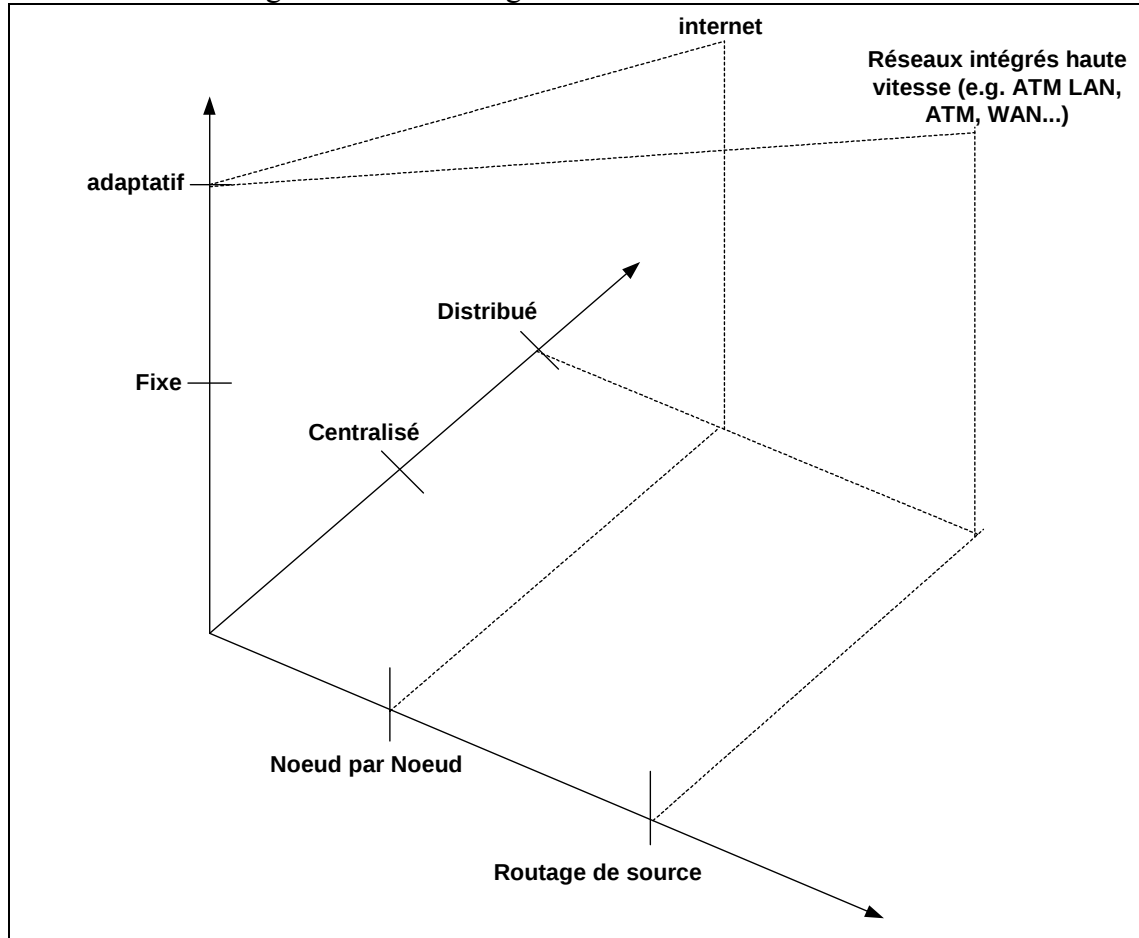
- La qualité de service que l'on est en mesure d'offrir à une connexion est directement liée au choix du chemin. Le calcul de route doit prendre en compte les différentes contraintes imposées par la connexion (débit, taux de perte), ces paramètres peuvent en effet varier en fonction des liens empruntés. Il faut donc mettre en oeuvre un algorithme de routage qui a pour rôle de trouver le meilleur chemin possible entre la source et le destinataire pour satisfaire les différents critères de qualité imposés. Il appartient tout d'abord à l'algorithme de réservation de bande, utilisé dans l'algorithme de routage, de déterminer la bande à allouer pour satisfaire ces contraintes. L'algorithme de routage utilise alors ces différentes informations dans une fonction de coût servant au calcul du chemin optimal
- Le calcul se doit également de répartir de la manière la plus homogène possible les ressources du réseau. Il faut éviter qu'en régime permanent certains liens soient surchargés, laissant d'autres inutilisés. On conçoit aisément qu'une répartition de la charge des ressources ne peut aboutir qu'à une amélioration des performances du réseau.
- Il faut enfin que le réseau puisse accepter le plus grand nombre possible d'appels par unité de temps, imposant à l'algorithme de routage de s'exécuter rapidement. Il faut donc rendre l'algorithme le plus simple possible, objectif souvent contradictoire avec les soucis d'optimalité désirés.

Classification des différents algorithmes

Du fait de la variété des objectifs qui sont visés, il existe plusieurs types d'algorithmes de routage. Ceux-ci peuvent correspondre à des politiques déterministes ou adaptatives selon qu'elles s'adaptent ou non aux variations du trafic et de topologie du réseau. D'autre part, les algorithmes de routage peuvent être centralisés (*Centralized*) si les chemins sont définis par un noeud particulier. Dans le cas contraire, l'algorithme de routage est réparti (*Distributed*) entre tous les

noeuds, ce qui est favorable du point de vue de la fiabilité, mais complique l'algorithme et rend plus difficile l'optimisation de l'acheminement des paquets. Le troisième paramètre à prendre en compte est le travail de routage effectué au niveau de chaque noeud, cela peut être le choix du prochain noeud ou l'indication de la route complète¹. Il existe enfin le routage par inondation ou aléatoire dans lequel le choix ne dépend pas de la topologie du réseau.

Classification des algorithmes de routage:



Ainsi un algorithme de routage résulte d'un ensemble de choix dans ces trois directions. Chaque algorithme peut donc être placé dans cet espace.

¹ Ce choix peut être lié au protocole de transmission utilisé: mode connecté ou non connecté.

Choix d'un algorithme de routage

Pour choisir un algorithme de routage, il faut tenir compte des contraintes matérielles de l'environnement. Il faut pour chaque paramètres constituant la stratégie de l'algorithme choisir parmi les possibilités qui sont offertes. La classification des algorithmes vus précédemment montre qu'il y a trois principaux choix à faire. Cette partie explique le fonctionnement de chaque famille d'algorithme, et évoque les variantes possibles.

Routage fixe et routage adaptatif

Routage fixe

La méthode de routage fixe définit les chemins que doivent suivre les différents paquets à partir des caractéristiques générales du réseau telles que la topologie et le trafic moyen escompté sur les différents canaux de transmission. Les règles d'acheminement sont en principe établies une fois pour toutes et visent à privilégier un critère de performance choisi par le concepteur. Dans la plupart des cas, l'optimisation du réseau vise à minimiser le temps moyen de traversée par les paquets. La méthode du routage fixe se traduit au niveau de chaque noeud par une table de routage dont la consultation permet au noeud de déterminer sur quelle voie de sortie il doit transmettre un paquet qu'il vient de recevoir, afin de l'acheminer dans les meilleures conditions possibles vers sa destination. La méthode du routage fixe est donc simple dans son principe, puisque le seul traitement effectué au niveau des noeuds se résume à la gestion des files d'attentes des voies et à des consultations de tables. Par conséquent, l'algorithme d'optimisation des chemins n'est mis en oeuvre qu'au moment de la conception du réseau. En pratique, les tables de routage ne peuvent pas être complètement fixes car la topologie du réseau peut changer par suite de défaillances d'équipements ou à cause de l'introduction de nouveaux utilisateurs. Le routage fixe est donc une méthode où les tables de routage ne sont que rarement remise à jour, par opposition au routage adaptatif, avec lequel les tables de routage sont corrigées en permanence, en fonction par exemple de l'état instantané du trafic. On voit que la méthode de routage fixe vise une optimisation globale à long terme, alors que la méthode adaptative a pour ambition de satisfaire à tout instant à un ou plusieurs critères d'optimalité.

Exemple de table de routage fixe pour le réseau suivant:

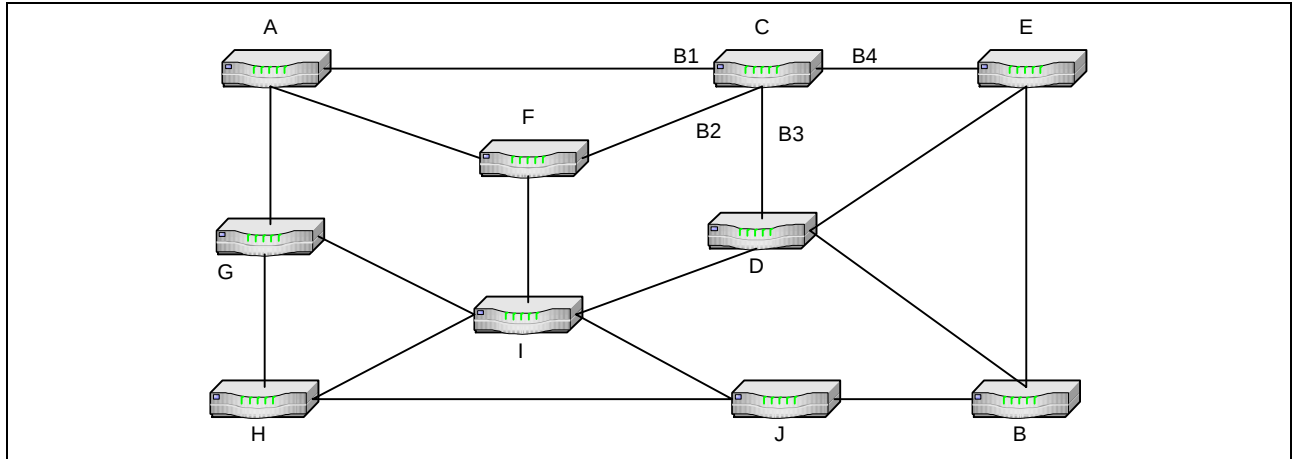


Table de routage du nœud C:

Adresse destination	Adresse du prochain nœud
A	A
B	D
C	-
D	D
E	E
F	F
G	A
H	A
I	F
J	F

Routage adaptatif

Bien que le routage fixe soit plus simple à mettre en oeuvre et à valider, dans un souci d'efficacité on s'oriente de plus en plus vers des méthodes de routage adaptatif. Le routage adaptatif est une solution nécessitant la mise à jour régulière des tables de routage en fonction des changements de topologie ou de charge du réseau. Par la suite, nous nous intéresserons exclusivement au routage adaptatif. Il est entendu que certaines des définitions qui seront données peuvent s'appliquer au routage fixe.

Sélection de la route

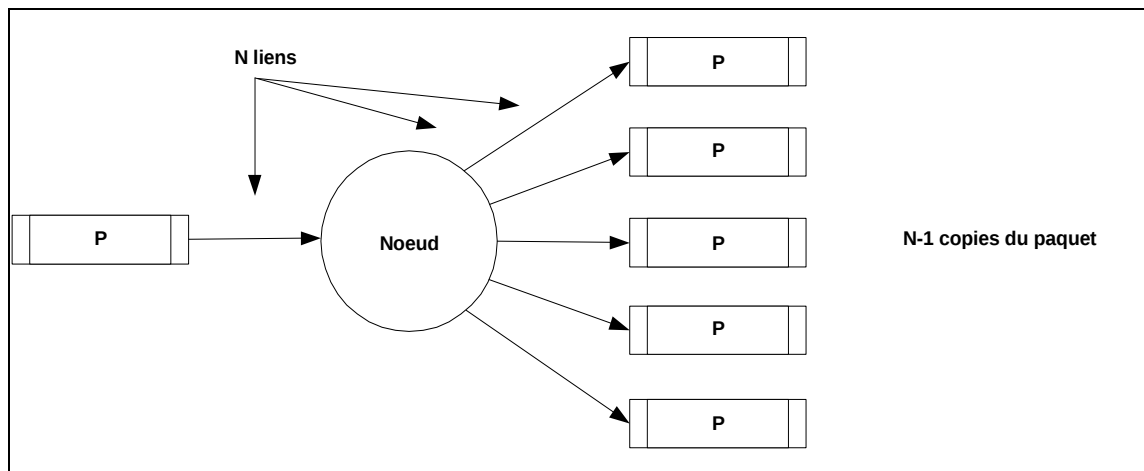
Routage Aléatoire

Les principes de routage par inondation sont difficilement classifiables car ils ne tiennent pas compte des caractéristiques du réseau pour effectuer le routage.

Acheminement par inondation

La technique d'acheminement par inondation (*Flooding*) est basée sur un principe simple consistant pour chaque noeud à retransmettre le paquet reçu sur toutes les voies de sortie du noeud, à l'exception de la voie d'arrivée. Un noeud connecté à N noeuds retransmet donc $N-1$ répliques du paquet qu'il vient de recevoir.

Principe du Flooding:



On voit que la méthode par inondation assure l'arrivée d'au moins une copie du paquet au destinataire pourvu qu'il existe au moins un chemin entre celui-ci et la source du paquet. Cette garantie d'acheminement est assurée même si la topologie change, à la suite par exemple d'une défaillance de certains composants du réseau. La méthode de l'inondation permet donc une grande robustesse dans l'exploitation du réseau, ce qui explique que ces applications principales soient du domaine militaire. D'autre part, comme toutes les liaisons possibles entre le noeud d'origine et le noeud destinataire sont essayées de façon exhaustive, la méthode assure qu'au moins une des répliques du paquet atteindra le destinataire par le chemin le plus court, et donc avec un délai minimum si le réseau est peu chargé. La technique par inondation présente également l'avantage d'être très simple à implanter, puisque l'acheminement des messages ne nécessite ni tables de routage ni même de connaître la position géographique du destinataire dans le réseau. En effet pour que le paquet parvienne à son destinataire, il suffit simplement que celui-ci soit capable de reconnaître que le message lui est adressé. Le prix à payer pour ces qualités de simplicité, de robustesse, et de rapidité d'acheminement est évidemment une mauvaise utilisation

des ressources du réseau et une tendance marquée à la congestion. En effet, puisque chaque noeud produit en général plusieurs répliques du même paquet, le nombre total des répliques en transit dans le réseau croît très rapidement en fonction du nombre de noeuds traversés. Cela explique pourquoi, en dehors de quelques cas particuliers, cette technique est peu employée. Néanmoins, ce principe peut être utilisé de manière efficace pour la transmission de routage dans le cas de certaines techniques adaptatives.

Acheminement aléatoire

La technique de l'acheminement aléatoire (Random Routing, Stochastic Routing) partage avec la méthode de l'inondation la caractéristique de ne pas nécessiter que les noeuds connaissent la structure du réseau ou l'état du trafic pour prendre la décision du routage à leur niveau. Toutefois, les noeuds évitent ici d'envoyer systématiquement sur toutes les voies de sortie des répliques des paquets qu'ils reçoivent, afin de ne pas produire un trafic fantôme trop important. La méthode de routage aléatoire consiste à émettre une ou plusieurs répliques du paquet reçu sur des voies de sortie qui sont choisies soit au hasard, soit en fonction d'une information sur la direction générale suivie par le paquet. Dans ce dernier cas, la méthode de routage est appelée inondation sélective (*Selective Flooding*). Avec la méthode d'acheminement aléatoire la plus simple, chaque noeud retransmet le paquet reçu sur l'une des voies de sortie choisie au hasard. Avec un réseau convenablement maillé, le paquet finit toujours par arriver au destinataire, même si le chemin qu'il doit suivre est compliqué. On voit que cette technique conduit à une réalisation très simple au niveau des noeuds, et qu'elle évite la saturation causée par la duplication des paquets. En contrepartie de ces avantages, le délai d'acheminement est en moyenne plus long qu'avec des techniques déterministes classiques, puisque les paquets suivent en général des routes tortueuses au lieu de prendre le chemin le plus direct vers la destination.

Noeud après noeud

Le routage noeud après noeud (*Hop by Hop*) consiste à ne donner que l'adresse du prochain noeud pour accéder au destinataire. Le résultat d'une table de routage reposant sur ce principe est donné pour le routage fixe. Ainsi à chaque noeud l'algorithme renvoie l'adresse permettant de se rapprocher du destinataire. Cette méthode est particulièrement adaptée aux protocoles basés sur le mode non connecté.

Routage de source

Le routage de source est la détermination complète de la route par la source. En effet, le premier commutateur décide de l'ensemble des noeuds à prendre pour accéder au destinataire. Pour permettre ce calcul, le noeud doit posséder les caractéristiques et la topologie de l'ensemble du réseau. Cette méthode est particulièrement utile pour les réseaux à haut débit ou l'on souhaite offrir un maximum de connections par unité de temps. Dans ces conditions, il ne faut pas que les performances soient détériorées par un temps de calcul de route trop long. Pour que ce routage

soit efficace, il est indispensable que les données sur le réseau soient parfaitement à jour, pour cette raison cette technique est plutôt réservée au routage adaptatif qu'au routage fixe.

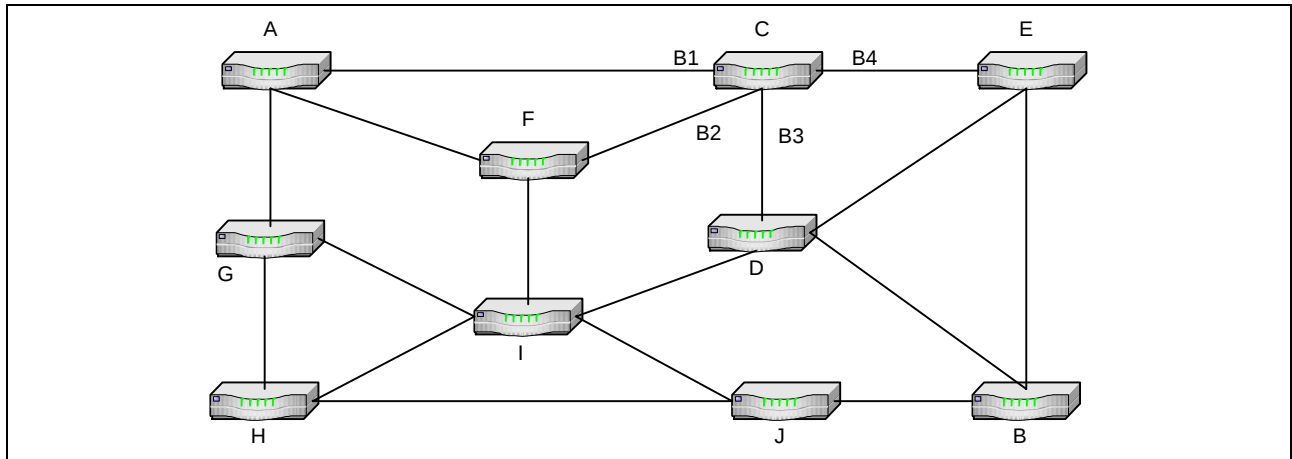


Table de routage du nœud C:

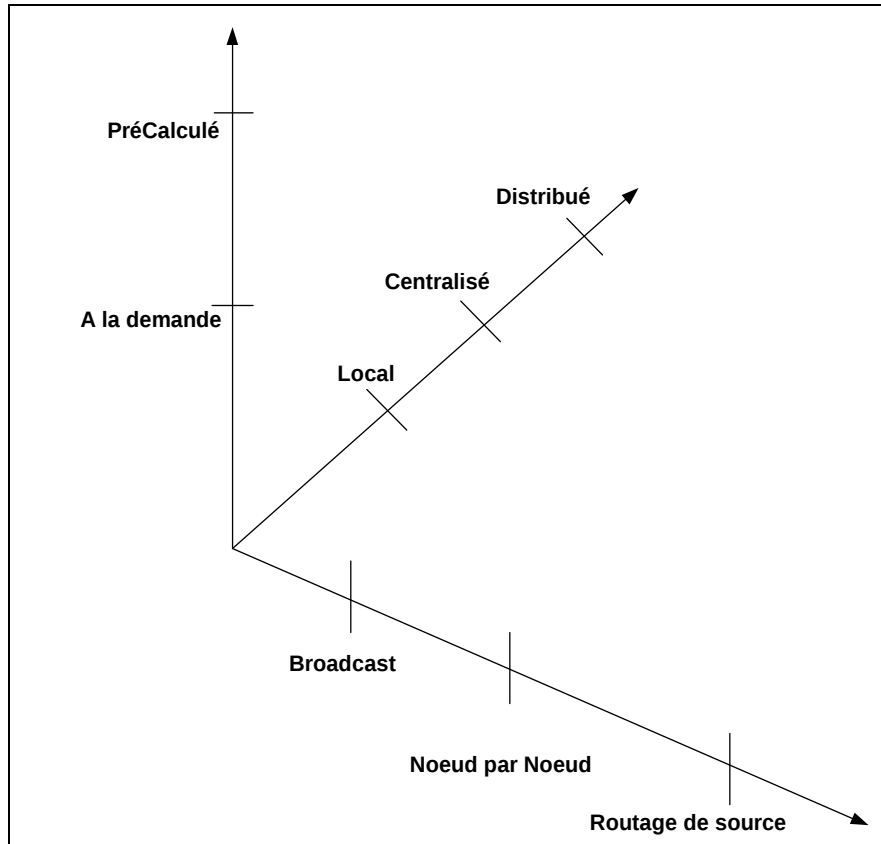
Adresse destination	Adresse du prochain nœud
A	A
B	D, B
C	-
D	D
E	E
F	F
G	A, G
H	A, G, H
I	F, I
J	F, I, J

Pour des réseaux de tailles raisonnables (typiquement quelques centaines de nœuds) on préférera le routage de source car il est plus rapide, le calcul de la route n'est effectué qu'une seule fois. Pour les autres réseaux, on choisira un routage adaptatif nœud par nœud pour ne pas surcharger en mémoire et en temps de calcul les nœuds du réseaux. Même si la contrainte prioritaire reste la taille du réseau, ce type de choix est aussi souvent lié au protocole utilisé qui impose des contraintes de mise en oeuvre trop importantes. On choisira, par exemple, plus facilement un routage nœud par nœud sur Internet, en effet la taille maximale des paquets étant fixe; effectuer un routage de source² imposerait à l'algorithme de faire de l'assemblage/désassemblage de données.

² qui implique l'encapsulation des données

Routages Adaptatifs

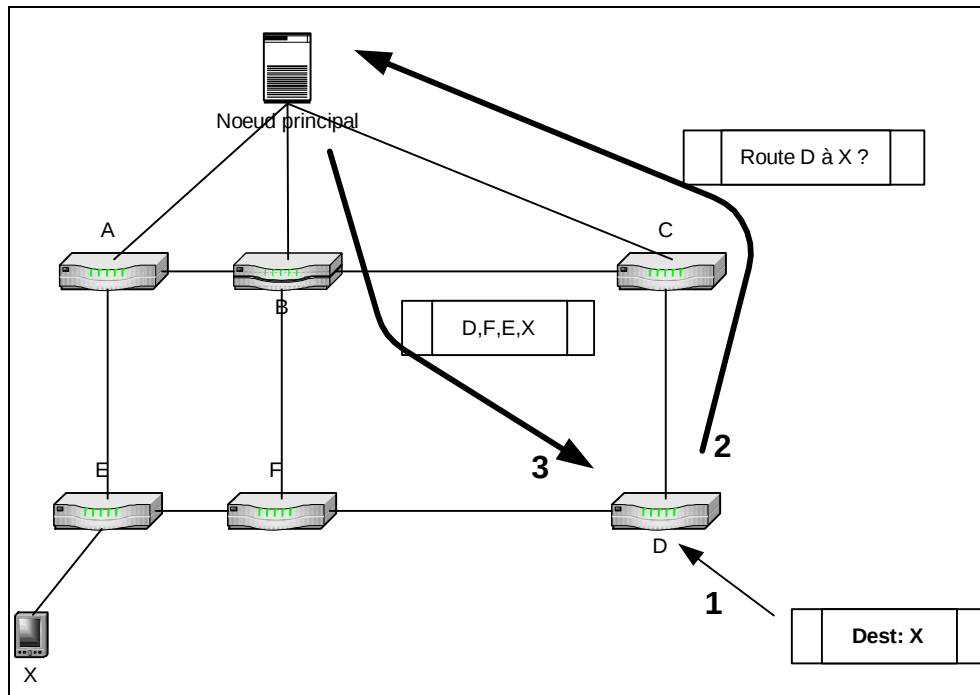
Classification des algorithmes de routage adaptatif:



Routage centralisé

Avec un routage centralisé, c'est un noeud qui possède toutes les informations sur l'état du réseau. Ce noeud est donc en mesure de calculer à chaque instant le chemin optimal entre deux noeuds. Ainsi tout noeud source désirant établir une connexion doit s'adresser au noeud "principal", ce qui augmente le temps pour calculer une route. De plus, il existe un problème de fiabilité important. En effet, si ce noeud de routage venait à être hors service, ou si un des liens le reliant au reste du réseau était coupé, il y aurait alors un impact sur le bon fonctionnement du réseau.

Principe du routage centralisé:



Routage local

Les méthodes de routage local et de routage réparti sont toutes deux basées sur l'établissement par chaque noeud de sa propre table de routage en fonction des informations qu'il peut glaner localement. La table de routage est par exemple dérivée de la table des délais estimés en choisissant pour chaque destination la ligne de sortie donnant le délai le plus court. Avec le routage local, chaque noeud établit sa table uniquement à partir des informations auxquelles il a accès directement, sans échanger d'informations avec ses voisins. Sous sa forme la plus simple, l'algorithme de routage local consiste pour chaque noeud à essayer de se débarrasser au plus vite du paquet qu'il vient de recevoir, en le plaçant dans la file d'attente la plus courte (*Hot Potato Algorithm*). L'algorithme local simple tend par son principe à faire circuler rapidement les paquets dans le réseau. Toutefois, comme les routes sont choisies de façon quelconque, la longueur des chemins est loin d'être optimum. Pour pallier ce problème, une méthode consiste à combiner le routage fixe à cette méthode.

Destination	Délais de sortie				Ligne choisie
	B1	B2	B3	B4	
A	0.2	0.3	0.4	0.5	B1
B	0.5	0.2	0.6	0.1	B4
D	0.2	0.3	0.1	0.6	B3

Routage réparti

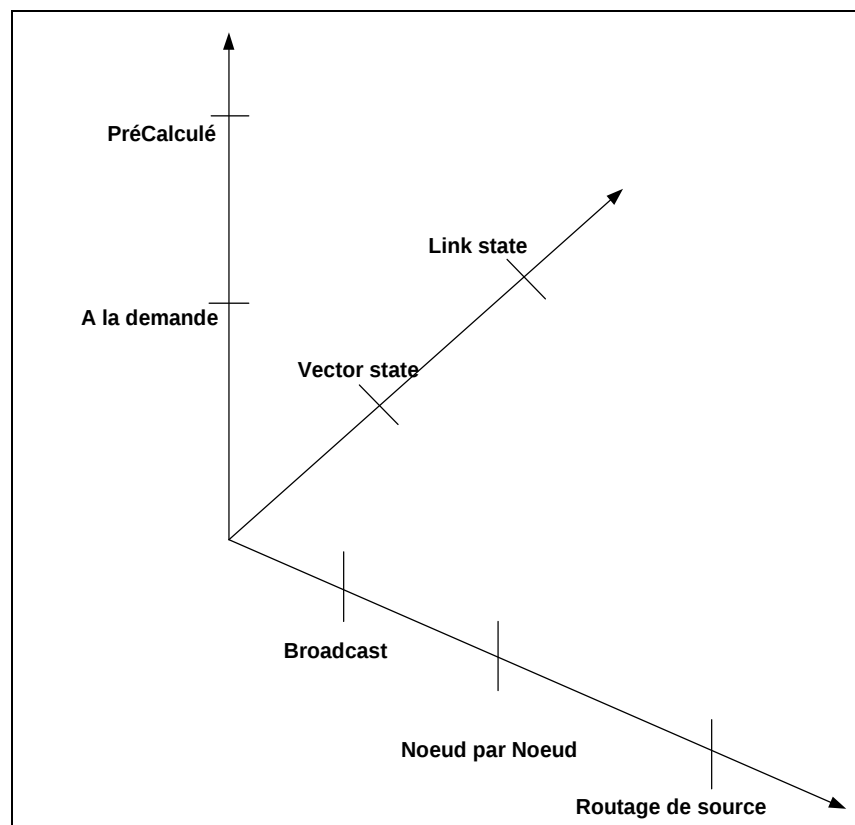
La technique du routage réparti est une méthode locale dans laquelle les stations voisines échangent des messages sur l'état du trafic et du réseau afin de mettre à jour leurs tables de routage. On conçoit donc que ce système adaptatif puisse, avec un certain délai, établir des tables

de routage qui minimisent les temps de transfert. C'est vers cette méthode que s'orientent les travaux, car avec l'apparition des réseaux haut débit, il est important de réagir rapidement au changement de topologie du réseau. Cette méthode reste fiable, et le dysfonctionnement d'un nœud ne compromet pas la bonne marche du système.

Routage adaptatif réparti (ou distribué)

Les performances des réseaux étant en constante progression, les algorithmes de routage utilisés doivent être de plus en plus aptes à répondre aux critères de bonnes gestion des ressources du réseau. Pour cette raison, les algorithmes de routage utilisés sont adaptatifs et distribués.

Classification des algorithmes de routage adaptatifs distribués:



Deux des principales caractéristiques du routage adaptatif réparti sont:

- la gestion d'une base de données contenant les caractéristiques du réseau (topologie, charge...)
- le calcul de route à partir de ces données

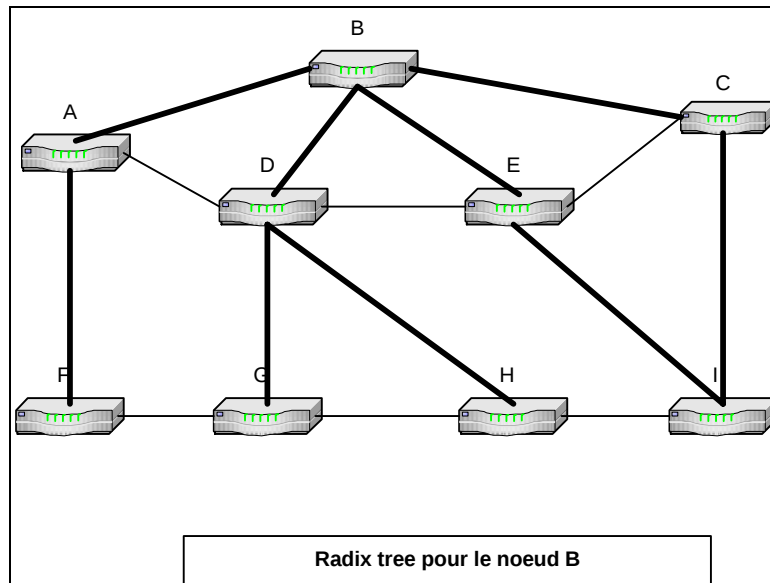
Pour permettre cela, il faut:

- Un protocole d'échanges d'information entre nœuds
- Un algorithme de calcul de route optimisé selon un critère donné

Calculer une route

On considère que l'algorithme de calcul de route dispose de toutes les informations nécessaires, c'est-à-dire qu'il connaît l'ensemble des connections du réseau et les poids associés à chacun de ces liens. Supposons que le calcul de route cherche à minimiser la valeur de la route. Le critère de sélection permet d'affecter des poids à chaque lien de manière à favoriser les liens de valeurs les plus faibles. Ces informations nous permettent de construire un graphe valué. Les algorithmes permettant de calculer un chemin de valeur minimum dans un graphe sont nombreux: l'algorithme de Bellman-Ford, de Dantzig, ou la méthode matricielle. Généralement, on ne cherche pas seulement un chemin de la source à un seul point, mais l'ensemble des chemins minimaux vers tous les autres points. Ce résultat est stocké dans le *optimal-path radix tree*. C'est un arbre unidirectionnel dont la racine est le noeud à partir duquel les distances sont calculées.

Exemple:



Chaque feuille représente une destination. Les prédécesseurs d'une feuille sont les noeuds par lesquels il faut passer pour accéder à la destination par un chemin optimum³. Le calcul de cet arbre est un problème classique de la théorie des graphes. On trouve de nombreuses sources décrivant les deux principaux types d'algorithmes résolvant ce problème. Le premier écrit par Dijkstra⁴ est en fait un cas plus simple du second Ford-Bellman-Moore⁵ car il suppose que la valeur de chaque arc est positive. La complexité de l'algorithme de Dijkstra est $\theta(n^2)$ alors que l'algorithme de Ford-Bellman est $\theta(n^3)$. Il existe néanmoins des techniques permettant de diminuer cette complexité.

Algorithme de Dijkstra:

³ Il est intéressant de constater que la complexité du calcul d'un chemin ou de tout l'arbre est la même avec l'algorithme de Bellman

⁴ label setting

⁵ label correcting

1. Soit V l'ensemble des nœuds du graphe, E l'ensemble des liens, T_s un ensemble de nœuds initialisé à $\{x_r\}$ où x_r est le nœud racine, E_s l'ensemble des liens du spanning tree initialisé à 0. $c(x_i, x_j) \in \mathbb{R}^+$ est la valeur de l'arc entre les nœuds x_i et x_j . $\forall t \in V - T_s$, $d(t) = \infty$ et $d(x_r) = 0$
2. $S = \{ (x_i, x_j) : x_i \in T_s \wedge x_j \in V - T_s \wedge (x_i, x_j) \in E_s \}$. Terminer si S est vide
3. Trouver $(x_k, x_l) \in S$ tel que $d(x_k) + c(x_k, x_l) \leq d(x_i) + c(x_i, x_j)$, $\forall (x_i, x_j) \in S$.
 $T_s = T_s \cup \{x_l\}$, $E_s = E_s \cup \{(x_k, x_l)\}$, $d(x_l) = d(x_k) + c(x_k, x_l)$. Recommencer en 2

Algorithme de Ford-Bellman-Moore :

4. Soit V l'ensemble des nœuds du graphe, E l'ensemble des liens, T_s un ensemble de nœuds initialisé à $\{x_r\}$ où x_r est le nœud racine, E_s l'ensemble des liens du spanning tree initialisé à 0. $c(x_i, x_j) \in \mathbb{R}$ est la valeur de l'arc entre les nœuds x_i et x_j . $\forall t \in V - T_s$, $d(t) = \infty$ et $d(x_r) = 0$
5. Trouver un arc $(x_i, x_j) \in E$, tel que $d(x_i) + c(x_i, x_j) \leq d(x_j)$. Terminer si il n'y en a pas
6. $T_s = T_s \cup \{x_j\}$, $E_s = E_s - \{(x_r, x_j)\}$, $E_s = E_s \cup \{(x_i, x_j)\}$, $d(x_j) = d(x_i) + c(x_i, x_j)$.
Recommencer en 2

Bien qu'une multitude de critères puissent être pris en compte pour le calcul du chemin le "plus court", généralement on utilise la longueur⁶ (Shortest Path) et la largeur⁷ (Widest Path). De façon générale, le critère de largeur est utilisé sur les réseaux LAN, et la longueur sur les réseaux WAN⁸.

⁶ La longueur est exprimée en nombre de nœuds entre la source et la destination

⁷ Bande passante

⁸ On utilise aussi la combinaison des deux critères largeur et longueur.

Diffusion de la topologie

L'objectif est de pouvoir diffuser une information (état de liens et/ou changement de topologie) à travers le réseau, pour cela il est nécessaire qu'une donnée soit diffusée vers tous les autres noeuds du réseau. Les différents principes sont les suivants:

- Le **Flooding** est la méthode la plus simple qui consiste à recopier le paquet reçu sur toutes les autres interfaces. Le problème majeur vient de la consommation de bande passante⁹.
- L'**adressage séparé** consiste à émettre une copie du même paquet pour chaque destinataire. Cette méthode ne peut être utilisée que pour de petits groupes en raison des besoins importants en bande passante et en stockage de données.
- L'**adressage multi destinataire** consiste à envoyer plusieurs copie d'un paquet permettant l'adressage multiple. Un paquet est envoyé sur chaque route partagée par des noeuds. La mise en oeuvre de cette méthode nécessite l'utilisation d'adresses de tailles différentes, ce qui est rarement supporté dans les réseaux.
- L'**acheminement le long d'un arbre** consiste à faire suivre les paquets le long d'un arbre. On utilise pour cela un *spanning tree* mis à jour dans chaque noeud à partir de la topologie du réseau.
- Le **Flooding contrôlé** consiste à diffuser l'information selon le principe du flooding. De façon à minimiser l'excès d'utilisation de bande passante, la diffusion de l'information est contrôlée par un protocole d'échange.

Les méthodes s'utilisant de manière quasi-systématique sont le *Flooding contrôlé* et le *Spanning Tree*. Le *Spanning Tree* est généralement utilisé sur les réseaux WAN, pour permettre une diffusion rapide de l'information¹⁰ sans surcharger le trafic sur les liens. Le *Flooding contrôlé* est généralement utilisé dans les réseaux LAN; il permet de diffuser l'information de manière relativement rapide étant donné que les temps de propagation sont bref et le nombre de noeuds est limité.

Application

Le principe de diffusion d'information étant choisi, il convient maintenant de savoir comment ce principe va être utilisé et surtout qu'elles sont les données que l'on va transmettre. A ce jour, les deux principaux protocoles sont le *distance vector route* et le *link state*.

Distance Vector

Cette approche est la plus ancienne, elle est basée sur l'algorithme de Ford-Fulkerson . Cette méthode a été largement implémentée dans des solutions telles que RIP¹¹, BGP¹² ou encore

⁹ voir Acheminement par inondation

¹⁰ Si la diffusion d'information le long d'un arbre est rapide et simple, la construction de cet arbre est très complexe, particulièrement lorsqu'apparaissent des changements de topologie. Pour cela, cette technique n'est utilisée qu'en cas de nécessité absolue

¹¹ Routing Information Protocol

EGP¹³. L'idée principale est que chaque noeud connaît la distance (valeur du lien) vers les noeuds adjacents et distribue cette information à tous ses voisins. Les informations sur les distances sont stockées si elles sont meilleures que celles déjà connues. Un calcul et/ou une distribution sont effectués quand il y a un changement dans le coût ou la découverte d'un lien. Le principal défaut de cet algorithme, malgré sa simplicité, est la lenteur de convergence. De plus les informations envoyées ne sont pas sélectionnées, en effet toute la table de routage est transmise à chaque fois. Ce surplus d'informations engendre une dégradation des performances du réseau en raison de la bande passante utilisée pendant des périodes de temps relativement longues (lenteur de convergence).

Link State

La méthode du vecteur de distance est de plus en plus délaissée au profit de la solution basée sur le poids des liens. Cet algorithme est le plus récent et constitue la base de nombreux algorithmes tel que OSPF¹⁴ et IS-IS¹⁵. Bien que le principe de base soit simple -elle consiste à distribuer la topologie à tous les noeuds de manière à ce que chaque noeud calcule sa table de routage- son implémentation est très complexe. Le calcul de route est facile à comprendre et est basé sur l'algorithme de Dijkstra. La distribution de la topologie est basée sur un flooding contrôlé de LSUs¹⁶, numérotés et envoyés selon des règles d'ordonnancement sophistiquées pour permettre au système de converger. Il est nécessaire de connaître ses voisins pour former les liens permettant les échanges d'informations. Il existe de plus des améliorations permettant de limiter l'utilisation du support de communication par l'algorithme de diffusion.

¹² Border Gateway Protocol

¹³ Exterior Gateway Protocol

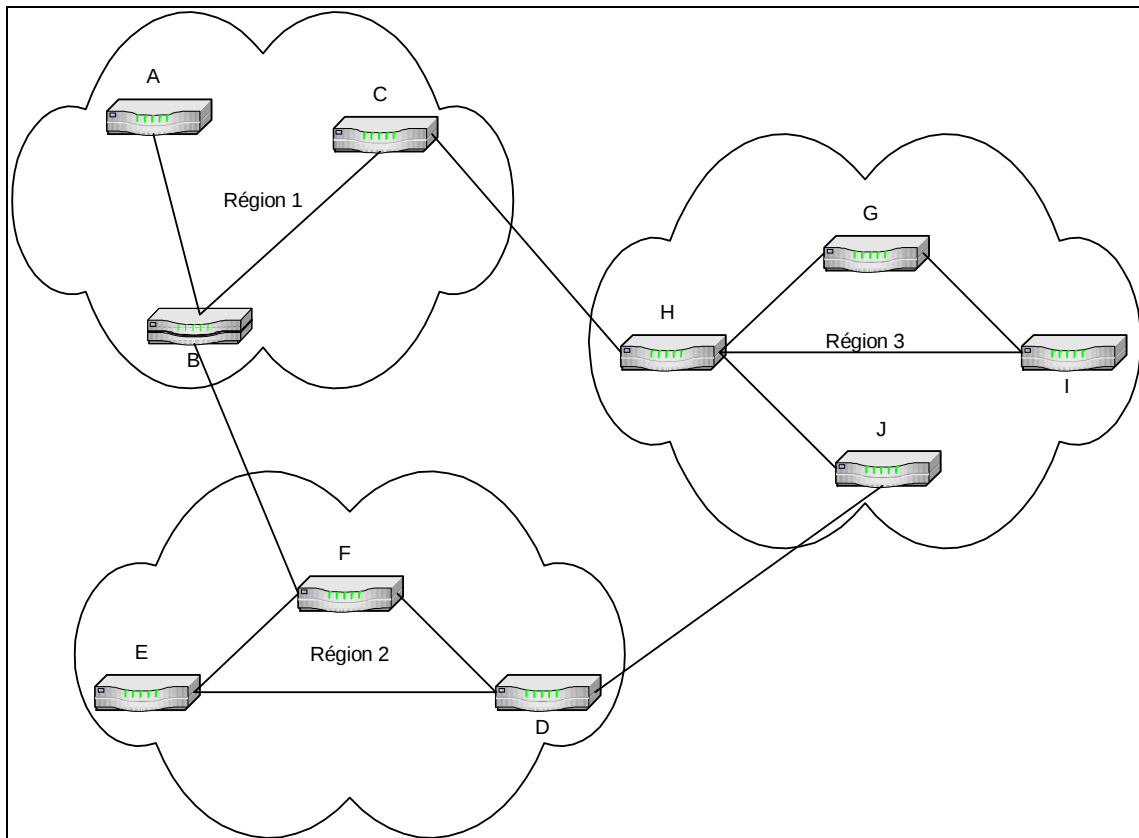
¹⁴ Open Shortest Path First, il fonctionne sur le protocole IP (Internet Protocol)

¹⁵ Intermediate System-Intermediate System, pour le modèle OSI

¹⁶ Link State Update

Routage hiérarchique

Avec les méthodes de routage non aléatoires, chaque noeud doit disposer d'une table de routage complète comportant la liste de tous les noeuds du réseau et l'ensemble des liens entre ces noeuds. De plus, lorsque le routage est adaptatif, les noeuds doivent échanger périodiquement des messages destinés à remettre à jour les tables de routage. La dimension des tables de routage et l'importance du trafic de service croissent donc rapidement avec le nombre de noeuds et deviennent inacceptables lorsque le réseau comporte plusieurs centaines de noeuds. Pour résoudre ce problème, il faut alors faire appel à une organisation hiérarchique du réseau qui regroupe les noeuds en grappes reliées entre elles par un réseau de niveau supérieur. On voit que cette méthode réduit radicalement la taille des tables de routage. En effet, celles-ci ne prennent plus en compte que les noeuds d'une région puisque tout paquet adressé à un noeud d'une autre région doit nécessairement transiter par le noeud faisant la liaison avec la région éloignée. Le prix à payer pour cette simplification du routage est évidemment une moins bonne optimisation du chemin parcouru par les paquets, puisque ceux-ci doivent passer par des points de passage obligés.



Conclusion

Les progrès techniques dans le domaine des réseaux permettent d'abandonner peu à peu certaines méthodes au profit des plus performantes. Pour cette raison, on ne parle maintenant plus que d'algorithmes de routage adaptatifs. On préférera de plus que l'algorithme soit distribué pour plus de sécurité. Pour ce type d'algorithme de routage, Le tableau suivant donne les choix que l'on a faire pour ce type d'algorithme et les critères généralement utilisés pour justifier chaque choix.

- Selon la taille du réseau on choisira plutôt:
 - o Routage de source: pour de petits réseaux.
 - o Routage noeud par noeud: pour des réseaux plus importants.
- La diffusion de l'information se fera par:
 - o Flooding contrôlé pour les LANs.
 - o Spanning Tree pour les WANs.
- Le critère de choix du meilleur chemin sera:
 - o La longueur (largeur) pour les WANs.
 - o La largeur pour les LANs.

Néanmoins, ces possibilités risquent à l'avenir d'être beaucoup plus nombreuses et le choix ne se fera plus sur un mais sur un ensemble de critères.

Ce tableau est simplifié mais il présente les choix généralement fait des les algorithmes existants ou en cours de recherche.

Nicolas JEAN

<http://www.nicolasjean.com>

Salemioche.net : création de site web pour les débutants

Nikozen : [hébergement professionnel](#) – [création site internet](#)

Glaces.org : recettes de glaces et sorbets

[Shopping Relax](#) : guide achat en ligne

[IP relax](#) : protocole http, smtp, pop, imap, irc, ftp, mime...