

Routage



et



fourmis



Sommaire

Introduction

I. Notions de routage

- a) Technologies actuelles
- b) Avantages et désavantages

II. Routage et fourmis

- a) Principe et avantages
- b) Structure du simulateur

III. Implémentation

- a) Présentation
- b) Détails

Conclusion

Bibliographie

Introduction

Depuis quelques années, le "haut-débit" s'est imposé dans un très grand nombre de foyers et les fournisseurs d'accès étendent sans cesse la gamme de services proposés. Du simple accès à internet à la télévision numérique en passant par la téléphonie, les besoins techniques ont rapidement évolués, tant en terme de quantité de données à acheminer qu'en terme de temps de réponse.

Les technologies réseau dites très haut débit, telles que le VDSL (Very high bit-rate Digital Subscriber Line) ou la FTTH (Fiber to the Home, fibre optique à domicile) sont en plein déploiement dans les grandes agglomérations, tandis que la télévision haute définition ou la VoD (Vidéo à la demande) sont les nouvelles coqueluches des consommateurs, alors que les joueurs réclament eux des temps de réponse toujours plus faibles.

Pour faire transiter correctement et efficacement toutes ces données transmises sous forme de paquets, il est impératif de minimiser le nombre de nœuds par lesquels ils passent, tout en optimisant la capacité du réseau et en gardant une certaine adaptabilité en fonction de l'état du réseau.

Les différents aspects du temps s'entremêlent donc au sein de ce problème : il s'agit d'optimiser en un temps minimal les temps de réponse et de transfert. Nous nous intéresserons donc aux solutions de routage actuellement utilisées, avant d'étudier une méthode de routage imitant le comportement social des fourmis.

I. Notions de routage

Le routage est l'ensemble des opérations visant à acheminer des données, sous forme de paquet, dans un réseau informatique complexe, dont les différents nœuds n'en connaissent pas la géométrie exacte. Un saut correspond au passage d'un paquet d'un nœud à un autre.

a) Technologies actuelles

- Le BGP, ou Border Gateway Protocol (Protocole de routeurs frontaliers), est le protocole de routage utilisé pour acheminer les paquets entre Systèmes Autonomes (ou AS, pour Autonomous System) sur Internet.

Ces AS sont en général la propriété de prestataires de services qui échangent d'importantes quantités de données, tels que les fournisseurs d'accès ou les hébergeurs de sites internet. Ils sont reliés entre eux par des liens de grande capacité, de 100 mégabits par seconde à plusieurs dizaines de gigabits par seconde. Il est donc crucial pour les AS de disposer d'un routage sans faille et, en raison des différents accords et partenariats commerciaux, configurable.

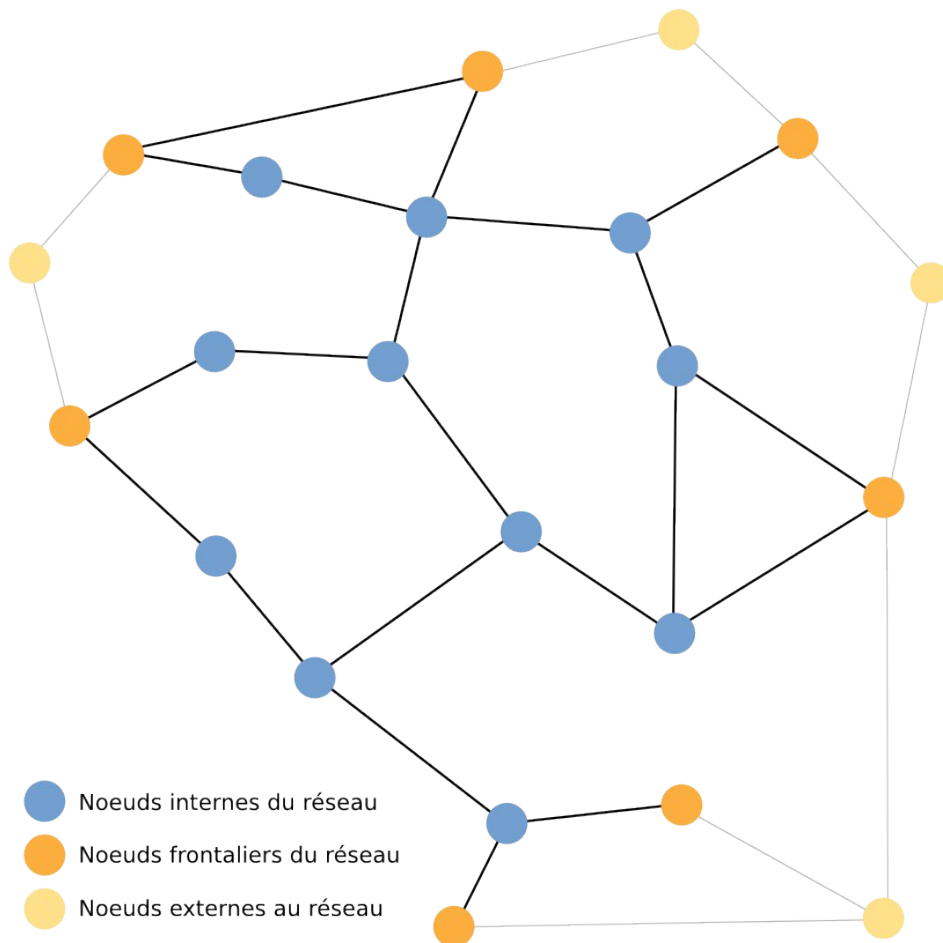
BPG propose ainsi un routage statique, où les chemins empruntés par les paquets sortants de l'AS sont définis manuellement dans chacun des routeurs. Les différents routeurs de l'AS communiquent entre eux et échangent leurs informations sur la géométrie du réseau interne afin de déterminer le point de sortie du réseau l'extérieur.

- Le protocole OSPF, pour Open Shortest Path First (Chemin ouvert le plus court en premier), est, contrairement à BGP, un IGP (Interior Gateway Protocol, ou Protocole de routeurs intérieurs). Il est utilisé pour router les paquets au sein d'un AS.

Chaque routeur du réseau envoie périodiquement à ses voisins l'état du réseau tel qu'il le voit, incluant des informations sur l'état des différents liens (le protocole est dit de type Link – State).

Les voisins mettent à jour leurs propres tables de routage si les informations reçues sont plus récentes que celles qu'ils avaient déjà, puis répètent le processus et envoient leurs informations. Ceci permet une certaine flexibilité vis à vis de la charge du réseau, et une certaine résistance aux pannes.

Le routage est ici dynamique, et s'appuie sur une variante de l'algorithme de Dijkstra, mathématicien et informaticien néerlandais de renommée mondiale. À partir des données collectées, chaque routeur construit un graphe du réseau tel qu'il le connaît, en utilisant comme poids l'évaluation qu'il a du lien considéré. Il applique ensuite l'algorithme afin de déterminer vers quel nœud les paquets doivent être envoyés au prochain saut, et ce pour chacune des destinations.



Exemple de réseau, le routage entre les nœuds internes et les nœuds frontaliers est effectué suivant une méthode similaire à OSPF, tandis que le routage entre deux nœuds externes *via* le réseau est géré par une méthode de type BGP

b) Avantages et inconvénients

Ces deux protocoles, les plus utilisés (et de loin) pour le routage des différents réseaux constituant Internet, semblent satisfaire une majorité des impératifs associés au problème du routage : le BGP est configurable et, de part sa simplicité, apte à router de grandes quantités de données, tandis que l'OSPF propose un routage dynamique évoluant selon l'état du réseau. Une combinaison de ces deux protocoles permet ainsi de router des paquets à travers des réseaux et ensembles de réseaux complexes appartenant à de multiples entités.

Ils souffrent cependant de faiblesses notables s'il faut garantir un routage optimal à chaque instant : le BGP n'est absolument pas adaptatif par rapport à l'état du réseau, tandis que l'OSPF souffre d'un délai non négligeable pour la mise à jour des tables en raison du temps de latence entre l'émission de données nouvelles en un point et leur réception en un autre. Enfin, ces solutions ne prévoient pas la possibilité de router des paquets ayant la même destination par deux routes différentes afin de minimiser l'encombrement des liens en exploitant au mieux la géométrie du réseau.

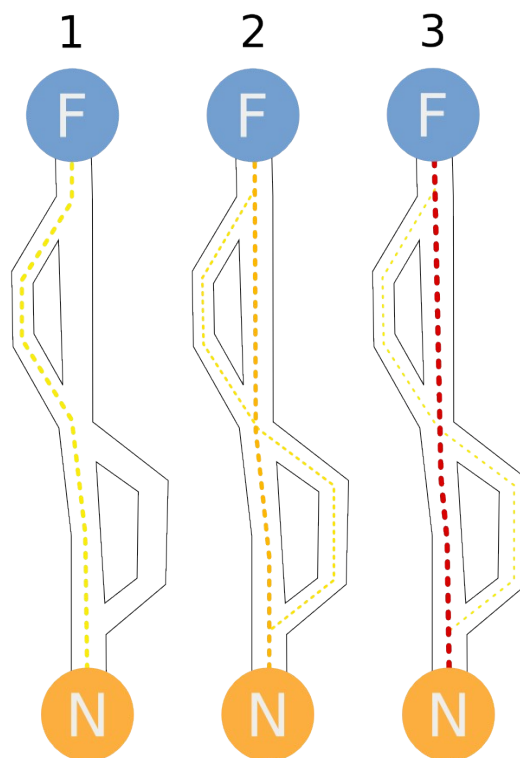
II. Routage et fourmis

a) Principe et avantages

Même s'il n'y a a priori que peu de rapport entre des insectes et un réseau informatique, si ce n'est dans le cadre de "bugs", imiter le comportement social des fourmis s'avère pourtant être une idée plus qu'à propos.

En effet, les fourmis, lorsqu'elles voyagent entre leur fourmilière et, par exemple, une source de nourriture, déposent sur leur passage des phéromones. Ainsi les premières fourmis se dirigeant vers la nourriture choisiront à chaque croisement une direction au hasard jusqu'à parvenir à destination. Elles reviennent ensuite à la fourmilière en déposant de nouvelles phéromones d'autant plus vite que le chemin est bon. Les fourmis suivantes emprunteront ensuite de préférence le chemin recouvert du plus de phéromones, et y déposeront leurs phéromones.

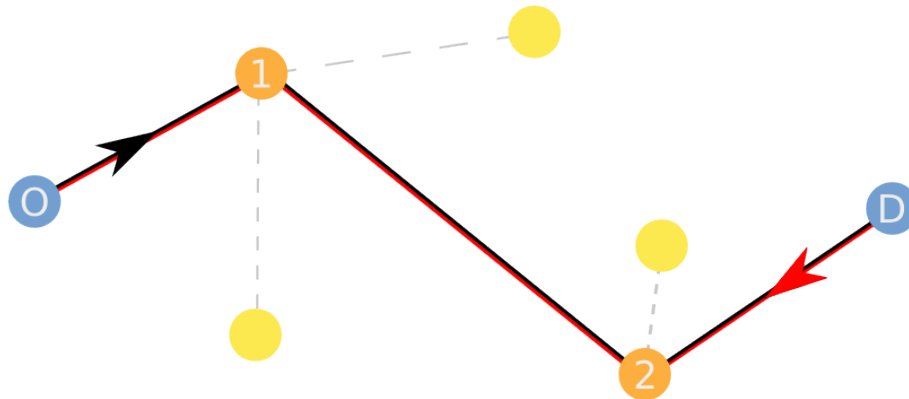
Ainsi au bout d'un certain temps le chemin optimal est recouvert de beaucoup de phéromones tandis que les autres seront presque vierges (d'autant plus que les phéromones s'évaporent au fil du temps).



Évolution des phéromones
sur les chemins visités par les fourmis

Intuitivement, cette méthode converge indéniablement vers la solution optimale, et l'adapter au routage est assez naturel. L'équivalent des phéromones est une valeur probabiliste. Les tables de routages utilisées par cette méthode sont plus évoluées que les tables classiques, qui ne contiennent qu'un seul saut possible par destination. Ici, la table d'un nœud contient, pour chaque nœud du réseau (la destination) et pour chaque nœud adjacent une valeur probabiliste évaluant la qualité du choix de ce nœud adjacent comme saut suivant pour atteindre la destination.

Aller : la fourmi collecte des données en étant considérée comme un paquet normal afin de fournir une évaluation empirique de l'état du réseau



Retour : la fourmi emprunte le même chemin en sens inverse en étant prioritaire par rapport aux autres paquets et met à jour les tables de routage sur son passage

Chaque nœud du réseau envoie à intervalle de temps régulier un agent mobile, appelé fourmi pour mettre en évidence l'analogie, vers un autre nœud du réseau choisi aléatoirement. Cette fourmi va être traitée comme n'importe quel autre paquet et rejoint sa destination tout en collectant des données statistiques sur son itinéraire. Ensuite, un nouvel agent appelé fourmi-retour est créé et parcourt le chemin de la fourmi précédente en sens inverse en mettant à jour les probabilités dans les tables de routages des nœuds empruntés, jusqu'à revenir à son point de départ et disparaître.

<u>Colonies de Fourmis</u>	<u>Routage</u>
Fourmi	Agent mobile sous forme de paquet
Fourmilière	Nœud origine
Source de nourriture	Nœud destination
Phéromones	Données des tables de routage

Analogie entre le routage d'un réseau
et le comportement social des fourmis

b) Structure du simulateur

- Initialisation du réseau
- Boucle principale
 - Génération de nouveaux paquets à router
 - Réception et traitement des paquets arrivant en chacun des nœuds
 - Traitement des files d'attente en sortie des nœuds
 - Mise à jour des données de routage
 - ▲ Traitement des paquets de routage reçus
 - ▲ Envoi aléatoire de paquets de routage

III. Implémentation

a) Présentation

L'implémentation a été développée en Python afin de pouvoir proposer une interface graphique décente, construite en utilisant les bibliothèques externes GTK et Cairo. Elle permet ainsi une configuration aisée des différents paramètres (nombre de nœuds, capacité des liens) et propose plusieurs modes de génération de trafic :

- Génération aléatoire de paquets à chaque nœud du réseau, afin de tester le comportement global de la méthode de routage.
- Génération d'une grande quantité de paquets en un unique nœud du réseau, afin de tester la réactivité et la souplesse du routage.
- Génération de nouveaux paquets sur les routeurs à la frontière du réseau afin de simuler un réseau utilisé pour faire transiter des paquets entre plusieurs réseaux.

Elle inclut en outre une visualisation en direct de l'état du réseau *via* une représentation colorée du réseau, et donne accès à diverses données statistiques ainsi qu'au contenu des tables de routage.

b) Détails

- Représentation du réseau

Le réseau est constitué d'un ensemble de nœuds. Chaque nœud possède une liste de liens adjacents, une file d'attente de paquets entrant et une file d'attente de paquets sortants pour chaque lien adjacent. La capacité des files d'attente ainsi que la bande passante de chaque lien (c'est à dire la quantité de données pouvant entrer à chaque instant dans le lien) peuvent être configurés.

- Gestion temporelle du réseau

Le nombre d'itérations de la boucle principale est utilisé comme repère temporel pour mesurer la durée des trajets. Bien qu'entière et imprécise, c'est la seule donnée qui ne soit pas influencée par les performances de la machine ou par l'état du réseau, les quantités de données à traiter variant par exemple d'une itération à l'autre.

- Initialisation des tables de routage

Soit un nœud n , ayant N nœuds adjacents, et un nœud destination d . Si d fait partie des nœuds adjacents à n , on lui attribue une probabilité plus forte, par exemple $\frac{1}{N} + \frac{3}{2} \frac{N-2}{N^2}$, les autres nœuds recevant une probabilité de $\frac{1}{N} - \frac{3}{2} \frac{1}{N^2}$. Sinon, les probabilités sont initialisées de manière uniforme (chacune des probabilités vaut $\frac{1}{N}$).

- Représentation des fourmis

Les fourmis sont de simple paquets portant une pile

d'enregistrements contenant chaque nœud emprunté et la date du passage de la fourmi en ce nœud. Lors du trajet aller, elles subissent les mêmes contraintes que les autres paquets (routage identique, files d'attente commune) afin de collecter des données représentant au mieux l'état du réseau. Au retour, elles suivent l'itinéraire de l'aller en sens inverse et sont prioritaires au niveau des files d'attente afin d'accélérer la propagation des informations.

- Mise à jour des tables de routage

Lors du trajet retour, à chaque nœud visité, la table de routage est mise à jour en regard du nombre d'itérations de la boucle principale effectuées entre le premier passage en ce nœud et l'arrivée au nœud destination. La table de routage conserve des données statistiques sur les précédents trajets. Ainsi, avec i est le nœud visité, p le nœud par lequel la fourmi est arrivée en i , d la destination originale de la fourmi, c la durée mesurée du trajet entre i et d , l la dernière durée constatée du trajet entre i et d , et m la durée moyenne d'un trajet entre i et d , la probabilité de router un paquet allant en d vers le nœud p est modifiée de $q = \frac{(l-c)}{m^2}$. Si il y a N nœuds adjacents au nœud i , alors les probabilités des autres nœuds pour ce même trajet sont elles modifiées de $q' = \frac{-q}{(N-1)}$.

Conclusion

Le routage par colonies de fourmis est donc une solution ingénieuse et flexible qui permet de composer avec une majorité des impératifs relatifs au routage d'un réseau informatique. En reproduisant le comportement social des fourmis, il est ainsi possible de mettre en place un routage reflétant réellement l'état du réseau et réactif. Ainsi, si un lien est subitement surchargé, les trajets effectués vers la même direction par les autres liens permettront malgré tout d'adapter le routage.

Au niveau des performances, le coût de la méthode est minime : les paquets de routage sont à peu près aussi nombreux que ceux impliqués dans des solutions comme OSPF.

Toutefois, ce n'est bien entendu pas une solution idéale puisqu'elle ne permet par exemple pas un routage statique prenant en compte les accords commerciaux inévitables, et ne peut donc être utilisé pour le routage entre plusieurs réseaux.

Le routage par colonies de fourmis apparaît donc au final comme un bon candidat comme remplaçant à OSPF, parfaitement apte à router des données au sein d'un réseau avec un maximum de flexibilité et de réactivité.

Bibliographie

- Two ant colony algorithms for best-effort routing in datagram networks
Gianni Di Caro et Marco Dorigo, 1998
<http://www.idsia.ch/~gianni/Papers/PDCS98.ps.gz>
- Improved AntNet routing
Benjamin Bardn , 2001
<http://www.cnc.una.py/cms/invest/download.php?id=90701,77,1>
- Ant colony optimization
Wikipedia
http://en.wikipedia.org/wiki/Ant_colony_algorithm
- Routing
Wikipedia
<http://en.wikipedia.org/wiki/Routing>