Magistère STIC 2006–2007
Module *Réseaux* Feuille de TP n° 3

Calcul distribué d'un arbre recouvrant

Dans ce TP nous allons implanter un algorithme pour le calcul distribué d'un arbre recouvrant.

La topologie initiale d'un réseau est un graphe dans l'ensemble des nœuds est partitionné en un ensemble de *lan.* et un ensemble de *ponts.* Toute arrête est entre un pont et un la (c'est donc un graphe *biparti*), et tout pont est connecté à exactement deux lans (c'est une hypothèse simplificatrice). Le graphe est supposé connexe

On cherche à calculer un sous-ensemble des ponts, appelé des *ponts actifs*, tel que pour deux lans donnés il y exactement un chemin reliant ces deux lans et qui ne passe que par des lans et par des ponts actifs.

Tous les ponts ont un numéro distinct. Une idée importante de l'algorithme est qu'on va élire une racine qui est le pont avec le numéro le plus petit.

Dans l'algorithme, chacun des nœuds maintient l'information suivante :
- la racine supposée
- sa distance avec la racine supposée

Les ponts peuvent envoyer des messages en multi-diffusion sur les lans avec lesquels ils sont connexes. Les messages contiennent trois composantes :
- le numéro de l'émetteur
- la racine supposée par l'émetteur
- la distance de l'émetteur à sa racine supposée

Au début, chacun des nœud se suppose lui-même racine et envoie un message à cet effet. Puis, les nœuds se mettent à écouter les messages envoyés sur les lans avec lesquels ils sont connexes. Les nœuds vont modifier leur information locale en fonction des messages reçus, et dans ce cas envoyer eux-mêmes des messages qui reflètent leur mise-à-jour.

Plus exactement, supposons qu'un nœud a le numéro n, il pense que la racine est r et que sa distance avec r est d. Le nœud reçoit un message n', r', d'.
- Si $r' < r$ alors il a appris de l'existence d'un racine plus petite que celle qu'il connaît, et met à jour sa racine et sa distance
- Si $r = r'$ et $d' + 1 < d$ alors il a appris de l'existence d'un chemin plus court vers la racine, et met à jour sa distance.
- Si $r = r'$, $d = d'$, $n' \neq n$, et si le message est reçu sur un lan qui n'est pas sur son chemin vers r alors le nœud va devenir inactif.

Exercice 1
Copier les fichiers du répertoire `~rtreinen/reseaux/tp3`. Vous y trouverez
- un fichier exemple `in` qui contient une description d'un graphe
- des fichier `lexer.mli`, `lexer.mll` et `parser.mly` pour la lecture des fichiers décrivant des graphes
- un `Makefile`
- le fichier `main.ml` sur lequel cous aller travailler. Ce fichier contient une classe pour les lans et une classe pour les bridges (la notion des classes vient de la programmation par objets, et sera expliquée en TP). Vous compléter l'initialisation, et la méthode `tick` de la classe bridge.

Exercice 2
Étendre l'exercice précédent pour des ponst qui peuvent être allumés ou éteints (on peut allumer ou éteindre un pont par cliquant dessus dans l'interface graphique).