

## TP 2 : Construction d'abris couvrants dans les graphes arbitraires

### 1 Le simulateur *JBotSim*

On va utiliser la bibliothèque java "JBotSim" afin de simuler des algorithmes distribués sur des réseaux arbitraires. La bibliothèque JBotSim est un logiciel libre disponible sur le site suivant :

<http://jbotsim.sourceforge.net/>

La documentation de la bibliothèque peut être trouvée sur le site suivant :

<http://jbotsim.sourceforge.net/javadoc/>

#### Exercice

1. Téléchargez le package JAR *JBotSim* et le programme HelloWorld sur le site web. Suivez les instructions pour compiler et exécuter le programme sur votre ordinateur.

<http://jbotsim.sourceforge.net/examples/01-HelloWorld.html>

2. Allez voir les exemples au lien ci-dessous pour comprendre les fonctionnalités de *JBotSim*.

<http://jbotsim.sourceforge.net/examples/index.html>

### 2 Implémentation de la construction d'arbre couvrant

#### Exercice

1. Implémenter l'algorithme de construction d'arbre couvrant avec initiateur unique (voir pseudo-code ci-dessous).
  - (a) Étendez la classe Node en créant la classe SPTNode qui implémente l'algorithme de construction d'arbre couvrant. La classe SPTNode devra aussi implémenter les interfaces OnClock et OnMessage. On utilisera le constructeur de la classe et les méthode *OnClock()* (pour spécifier le comportement de l'initiateur) et *OnMessage()* (pour spécifier le comportement d'un nœud à la réception d'un message) afin d'implémenter l'algorithme d'arbre couvrant.
  - (b) On va considérer des réseaux dont les nœuds ont des identifiants uniques (UID). Il faut donc assigner à chaque nœud créé un identifiant unique. Pour cela, on va utiliser un compteur en variable globale.

```
public static int counter = 0;
```

Quand un nœud est construit, on va fixer son UID à la valeur courante du compteur et incrémenter celui-ci.

- (c) L'initiateur unique qu'on va appeler chef (leader en anglais) sera toujours le nœud ayant un UID égal à 10.
- (d) Dans le fichier Main, on va créer un type de réseau (topology) dont les nœuds sont de type SPTNode. On va créer un bouton qui va permettre de démarrer l'algorithme grace au code suivant :

```

public class Main {
    public static void main(String[] args)
    {
        JButton button = new JButton("Start");
        button.setBounds(0,0,100,50);
        button.addActionListener(new SPTstart());

        Clock.pause();
        Topology t = new Topology();
        Node.setModel("default", new SPTNode());
        JTopology jtp = (new JViewer(t)).getJTopology();
        jtp.add(button);
    }
}

public class SPTstart implements ActionListener{
    public void actionPerformed(ActionEvent arg0) {
        Clock.resume();
        //System.out.println("test");
    }
}

```

2. Exécutez le programme et créez un réseau arbitraire en cliquant sur l'écran pour créer des nœuds. Observer l'exécution de votre algorithme et vérifier que celui-ci calcule bien un arbre couvrant (les liens en rouges devraient correspondre à l'arbre couvrant).

Algorithme pour la construction d'arbre couvrant avec initiateur unique

#### **Code for Initiator (Leader) :**

Send(UID) to each neighbor;  
Set Parent to this node;

#### **Upon receiving a message Msg :**

```

if (State != LEADER and Parent == null) then
{
    Set Parent to the sender of Msg;    // <Msg.Source>
    Send(UID) to each neighbor except Parent;
    Set color of the link to Parent to "red";
}

```

Les fonctions utiles pour le TP :

- void send(Node dest, java.lang.Object content) dans Node
- void setColor(java.lang.String color) dans Node et Link
- void send(Node dest, java.lang.Object content) dans Node