

TME2 ANET: Datas Aggregation.

We consider a set of fixed sensors deployed in a forest. They are able to measure the temperature, and therefore to detect fires. These sensors are able to communicate with each other thanks to wireless communication.

The idea is to control the temperature of a whole forest. Each sensor controls an area of the forest. A special node, called the station collects the informations of the sensors in order to alert firemen in case of a fire.

The communications between the sensors or between the sensors and the station are done following the links of a spanning tree where the root is the station.

1 Creation of the Spanning Three

To create a spanning tree of a network in a distributed way, the root has to send a message to all of its neighbors. The first time a node receives a message it choses the sender of this message as parent and sends a message to all of its neighbors.

You have to create two classes that extend the class Node of JBotSim. A class StationNode and a class SensorNode. To create a spanning tree, for the class StationNode you have to override the method OnStart() and for the class SensorNode you have to override the method onMessage().

Each SensorNode possesses a reference to its parent and to its children.

The main class of your program is the following.

```
import jbotsim.Topology;
import jbotsim.ui.JViewer;

public class Main {
    public static void main(String[] args) {
        StationNode station = new StationNode();
        station.setSize(15);

        Topology tp = new Topology(600, 600);

        tp.setCommunicationRange(150);
        tp.addNode(200, 200, new TreeNode());
        tp.addNode(400, 200, new TreeNode());
        tp.addNode(430, 300, new TreeNode());
        tp.addNode(500, 200, new TreeNode());
        tp.addNode(500, 400, new TreeNode());
        tp.addNode(300, 300, new TreeNode());
        tp.addNode(200, 400, new TreeNode());
        tp.addNode(450, 500, new TreeNode());

        tp.addNode(300, 100, station);
        new JViewer(tp);
    }
}
```

Q1. Write the class `StationNode` and the class `SensorNode`.

2 Sensing Sensors

Q2. Add a variable of type `Integer` to the `SensorNode` and a method `sense()` without argument that updates the value of this variable. The value of this variable is randomly chosen between 0 and 255. Set the color of the node depending on the value of its variable.

Q3. Copy at the end of the main class the following code that calls the method `sense()` when the command “update values” is selected. Don’t forget to suppress the instruction “`new JViewer(tp);`” in the main class.

```
JTopology jtp = new JTopology(tp);
jtp.addCommand("Update values");
jtp.addCommandListener(new CommandListener() {
    public void onCommand(String command) {
        if (command.equals("Update values")) {
            for (Node node : tp.getNodes()) {
                if (node instanceof TreeNode) {
                    ((TreeNode)node).sense();
                }
            }
        }
    }
});
new JViewer(jtp);
```

3 Datas aggregation

A Naive Protocol

The `SensorNode` have to communicate the temperature they detect each time it is updated. To do so, each node forward its temperature to its parent. Each time a node receives a temperature it forwards it to its parent, until the `StationNode` receives the value.

Q4. Implement this protocol. Write traces to check your implementation.

Q5. How many messages are transmitted in the worst case? (the worst case is when the tree is a chain).

A Smart Protocol

We need to preserve the energetic consumption of the sensors. Therefore we need to use algorithm that send very few messages. The `StationNode` doesn’t need to know all the temperatures. Indeed, it is used to act in case of a fire in the forest. Therefore the `StationNode` can just be aware of the maximum temperature detected, if this temperature is higher than a certain threshold then it can ask for more informations from the `SensorNode`.

Each `SensorNode` waits to receive the temperature of all of its children before sending the maximum between these temperatures and its own temperature to its parent.

Q6. Implement the protocol where the `StationNode` only receives the maximum temperature of the sub-trees rooted by its children.

Q7. How many messages are transmitted in the worst case?