

Projet de pôle

Programmation système et orientation TSI

Organiser l'activité dans les réseaux de capteurs : le problème de couverture de surface

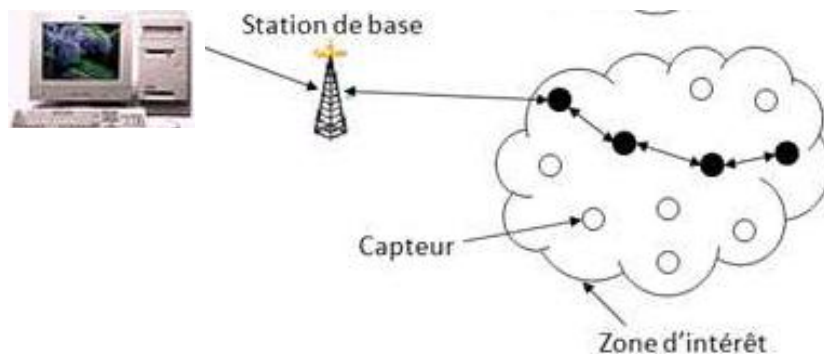
I. Introduction

Les **capteurs** sont employés pour relever la température, l'humidité, ou encore pour mesurer l'intensité lumineuse ou sonore. Récemment, ces capteurs se sont vus ajouter des capacités de communication et rendant possible l'accès à distance aux informations collectées. On parle alors de **capteur communicant**.

Lorsque deux capteurs sont **incapables de communiquer directement** (i.e. distance trop grande, environnement défavorable) et qu'**aucune infrastructure** n'est disponible, les communications transitent par plusieurs intermédiaires afin d'atteindre leur destinataire; on parle alors de communications **multi-sauts**.

Les réseaux de capteurs sont déployés sur une zone dans un but précis d'observation de leur environnement. Petits, voire minuscules, leurs ressources matérielles sont limitées (i.e. faibles capacités de stockage et de calcul). Pour assurer une observation permanente de la zone et garantir l'acheminement des données collectées via des communications multi-sauts en direction des **stations puits** (machines bénéficiant de capacités de traitement et de stockage en théorie illimitées), Les nœuds du réseau doivent s'organiser. Cette station puit sert aussi à activer les capteurs communicants parmi l'ensemble des capteurs de telle manière à couvrir la zone d'intérêt

Le critère utilisé ici est celui de la couverture de zone; les nœuds actifs doivent surveiller une zone aussi large que celle couverte par l'ensemble des nœuds et relaient les messages de collecte de données pendant que d'autres économisent leur énergie dans un mode passif.



Travail

1. Ecrire un fichier XML qui décrit l'ensemble du réseau avec l'ensemble capteurs-stations
2. Un programme moniteur qui se trouve dans la station de base (nœud puit) définit la topologie des stations à activer pour couvrir la zone d'intérêt en chargeant dynamiquement les classes de traitement sur ces stations
3. Si une des stations de la topologie choisie tombe en panne, le programme doit redéfinir une nouvelle topologie pour couvrir la zone d'intérêt, en arrêtant les programmes tournant sur les stations qui n'appartiennent pas à la nouvelle topologie, et activer les programmes sur les nouvelles stations
4. La communication entre stations se fera par :
 - a) Mémoire centrale (mémoire partagée) + sémaphores [**GI, JAVA**]
 - b) Pipe [**GI, JAVA**]
 - c) Socket [**TSI, C**]
5. Le programme moniteur contrôle le réseau en utilisant des signaux [**TSI, C**]
6. La récupération des données finales à partir du nœud «puits» se fera par clé USB [**TSI, C**]
7. La distance entre nœuds est assez grande, afin d'éviter l'altération des données qui circule sur le réseau, on procède par la modulation, l'échantillonnage et le codage du train de bits qui composent les messages qui circulent entre nœuds du réseau [**TSI, C**]
8. Les données sont stratégiques, pour les sécuriser, on utilise des procédures de cryptage [**TSI, C**]
9. En se basant sur votre cours de SysML modélisez votre système [**TSI**]

Remarque concernant les MSI

1. Comme les étudiants de MSI ne participent qu'au cours de programmation système, afin de valider ce module, soit :

1. Se contenter de la note de l'examen final de la matière ProgSys
2. Participer à une partie du projet mentionnée par [GI, JAVA]

2. Le choix doit se faire en début du semestre, et il est irrévocable