

Projet de fin des cours "Programmation en C" et "Réseaux"

Iryna Andriyanova et Tianxiao Liu

Octobre-novembre 2012

Résumé

Dans ce projet, il s'agit de concevoir et programmer un protocole de communication pour un réseau de capteurs. Ce protocole est une version simplifiée des protocoles du standard ZigBee, ZigBee étant un standard des communications sans fil, développé pour les terminaux périphériques (capteurs, contrôleurs, claviers sans fil,...)

Les connaissances, données aux cours "Programmation C" et "Réseaux" sont suffisantes pour réussir le projet.

Vous êtes invités à établir la communication entre les différents types de terminaux du réseau, notamment les terminaux-capteurs (jouant le rôle des "clients"), les terminaux-routeurs ("clients" ou "serveurs", en fonction de la direction de la communication) et le terminal-coordonateur (jouant le rôle du "serveur").

Le projet contient deux parties, la partie 1 et la partie 2. La partie 1 du projet consiste en développement du protocole de type UDP entre les capteurs, les routeurs et le coordinateur du réseau (voir Figure 1). La partie 2 consiste en gestion de la mobilité des capteurs. Les parties sont notées 14 points et 6 points chacune, faisant en somme 20 points.

Dans Section 1, vous trouverez la spécification générale du projet. Dans Section 2, les étapes du déroulement du projet sont données. Section 3 contient les critères de l'évaluation.

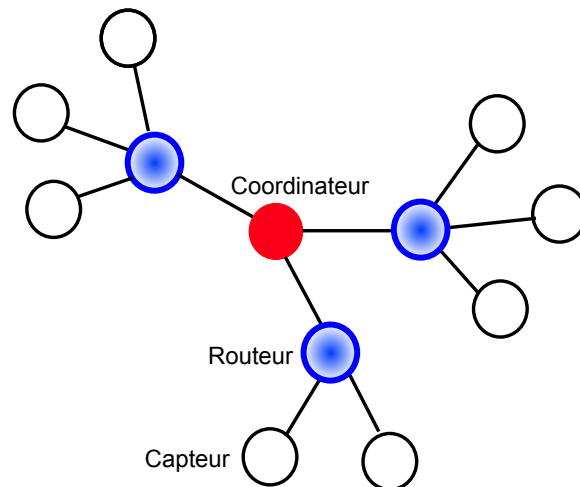


FIGURE 1 – Réseau de mesure de la température moyenne. Ici les capteurs de température sont connectés au noeud-coordonateur via trois noeuds-routeurs.

1 Spécification

Un exemple du réseau des capteurs est présenté sur Figure 1. Il consiste de :

- capteurs, prenant les mesures de température et les envoyant aux routeurs ;
- routeurs, qui reçoivent l'information des capteurs et la renvoient vers le coordinateur ;
- coordinateur, qui calcule la température moyenne mesurée par le réseau en se basant sur les mesures effectuées.

Pour simuler la mesure de la température, utilisez la fonction `mesure_temperature()`, disponible dans le fichier `projet_lpie.c`

1.1 Partie 1 : communication UDP (14 points)

Concevez et programmez le protocole de type UDP pour le réseau sur Figure 1, en utilisant le protocole UDP. A la soutenance du projet, vous devez présenter les codes C `capteur.c`, `routeur.c` et `coordinateur.c` qui assurent le fonctionnement suivant :

1. Deux possibilités d'adressage sont possibles :
 - L'identité du capteur est donnée par 16 derniers bits de son adresse IP, et cette adresse est unique dans le réseau. Le numéro du port est commun pour tous les noeuds.
 - ou
 - L'identité du capteur est donné par son numéro de port, qui est unique dans le réseau. L'adresse IP est commun à tous les capteurs (par exemple 127.0.0.1).
2. Le protocole IP utilisé est IPv4.
3. Les capteurs prennent les mesure de température et envoient les messages au routeur.
4. Le routeur collecte l'information de tous les capteurs disponibles et envoie un message au coordinateur.
5. Après avoir reçu les messages de tous les routeurs, le coordinateur calcule la température moyenne captée et l'affiche cette information. Si la température moyenne est plus grande que 50C, le message d'alarme est affiché.
6. La prise de température se passe périodiquement, chaque 2 minutes, et chaque valeur mesurée est envoyé vers le coordinateur. La communication doit être rapide (les transmissions via réseau sont finies avant la prochaine mesure).

1.2 Partie 2 : gestion de l'arrivée et du départ des capteurs (6 points)

Modifiez vos programmes pour les capteurs, routeurs et coordinateurs de manière à pouvoir effectuer les opérations suivantes :

1. Chaque noeud doit avoir une table de voisinage avec les adresses des ses voisins. La table initiale correspond à la topologie présentée sur Figure 1, mais elle peut être modifiée au cours de la communication.
2. Un nouveau capteur, arrivant au réseau, a une table de voisinage vide. Il peut essayer de se connecter a) au capteur, b) au routeur ou c) au coordinateur en envoyant un commande HELLO.

Pour simplifier, on suppose qu'on connaît l'adresse du noeud auquel on se connecte. Dans le vrai réseau ZigBee, le nouveau terminal diffuse simplement un message commun et attend la réponse des voisins.

Dans le cas a), le capteur doit recevoir un message d'erreur ERROR (un capteur ne peut pas se connecter au capteur). Dans les cas b) ou c) le capteur est autorisé à joindre le réseau ; on lui attribue une adresse parmi les adresses encore disponibles. Les tables de voisinage du routeur/coordonateur et du capteur sont modifiées.

3. L'arrivée des nouveaux routeurs ou coordonateurs n'est pas prévue.
4. Le noeud voulant quitter le réseau (par exemple, à cause de fin de la batterie), en informe ses voisins par la commande QUIT. Les noeuds voisins le prennent en compte et modifient leurs tables de voisinage.
5. Les capteurs non-connectés au réseau se mettent en veille et ne sont plus disponibles (l'exécution du code est terminée).

2 Déroulement du projet

Vous êtes libres de concevoir le protocole, à condition de respecter les points de la spécification ci-dessus. Nous vous proposons de procéder par des étapes suivantes :

1. Conception du protocole

Préparez le diagramme du protocole UDP capteur-routeur-coordonateur en mettant toutes les étapes et commandes à utiliser.

Décrivez votre choix d'adressage.

Décrivez quels messages seront envoyés par chacun de noeuds (une astuce : un routeur n'est pas obligé de faire suivre tous les messages des capteurs au coordonateur, mais il peut générer un message commun en se basant sur tous les messages venant des capteurs).

2. Programmation de la communication "1 capteur → 1 routeur" et "1 routeur → 1 coordonateur"

Créez les programmes `capteur.c`, `serveur1.c`, `serveur2.c` et `coordonateur.c` en se basant sur les programmes développés en TPs 2-3 des Réseaux.

3. Programmation des connexions simultanées, notamment "plusieurs capteurs → 1 routeur", "plusieurs routeurs → 1 coordonateur"

Modifiez les programmes `serveur1.c`, `serveur2.c` et `coordonateur.c` de façon à pouvoir gérer les connexions multiples (TPs 3-4 des Réseaux).

4. Programmation du passage des messages d'un capteur au coordonateur via un routeur

Mettez ensemble les codes `serveur1.c` et `serveur2.c` pour créer un seul `serveur.c`.

5. Programmation de la gestion de la mobilité

Ajoutez les options supplémentaires aux capteurs, routeurs et au coordonateur pour pouvoir gérer les messages HELLO et QUIT.

3 Evaluation du projet

Votre projet sera évaluée durant la soutenance, où vous présenterez le diagramme du protocole, le code C pour tous les noeuds (capteur, routeur et coordinateur), et ferez une démonstration des prises de mesure dans le réseau.

L'**évaluation du travail** va être effectuée par les critères suivants :

- Design du protocole : Expliquez vos choix faits lors de la conception du protocole à l'aide du datagramme. Par exemple, précisez quelle est votre stratégie de l'attribution des adresses dans le réseau, quels messages seront envoyés par les capteurs/routeurs etc.
- Fonctionnalité : les noeuds du réseau doivent envoyer, recevoir et afficher les messages correctement. Durant la soutenance du projet, vous ferez la démonstration du fonctionnement de votre code sur un PC.
- Qualité de programmation en C : vos programmes C doivent être claires et simples.