

TP1 IoT : Simulation d'un réseau de capteurs sans fil

Dans ces trois TPs, nous allons dans un premier temps utiliser le simulateur Cooja (TP1) pour mieux comprendre le fonctionnement des protocoles 802.15.4, 6LoWPAN et RPL. Dans un second temps, nous allons programmer des applications communicantes sur l'OS Contiki et les exécuter sur de vrais capteurs TelosB (TP2). Enfin, nous interrogerons directement les capteurs depuis un navigateur Web grâce au protocole COAP. Les objectifs sont donc les suivants :

- être capable de comprendre le fonctionnement des protocoles 802.15.4, 6LoWPAN et RPL
- être capable de développer un programme simple sous Contiki et de le déployer sur des capteurs
- être capable mettre en œuvre un service d'IoT grâce à COAP

Exercice 0 : Préparation de l'environnement

Installation de l'environnement de développement Contiki OS / TelosB

- Étape 1 - Télécharger contiki-2.7.zip: <http://sourceforge.net/projects/contiki/files/>
 - extraire l'archive dans votre home : /home/student
 - installer avec « sudo apt-get install » les paquets suivants :
 - ant
 - gcc-msp430
 - aller dans le répertoire ~/contiki-2.7/tools/cooja/
 - lancer Cooja avec la commande: ant run
- Étape 2 - Pour vérifier le bon fonctionnement de votre installation, suivre attentivement l'étape 3 « Run Contiki in simulation » du tutorial suivant :
 - <https://web.archive.org/web/20180818092938/http://contiki-os.org/start.html>
 - Lire la description détaillée fournie dans Cooja (colonne de droite) pour les différentes fenêtres du simulateur : Network, Timeline, Mote Output et Simulation Control.
 - A noter : Cooja simule les communications réseau mais utilise l'émulateur MSPSim pour émuler finement (au niveau des instructions) l'exécution d'un programme sur une plateforme basée sur un processeur MSP430.

Exercice 1 : 802.15.4

L'objectif de ce premier exercice est d'observer le fonctionnement de la couche radio/MAC.

- Question 1 - En sélectionnant un nœud et en dézoomant sur la fenêtre Network, vous pouvez observer trois zones (verte, grise et blanche). A quoi correspondent-elles ?
- Question 2 - Faites correspondre les paquets observés dans « Radio messages » avec ceux affichés dans les fenêtres « Mote output » et « Timeline ». Il suffit de faire un clic droit sur une trame et choisir une option dans le menu « Show in », ou sinon appuyer sur « espace » après avoir sélectionné une trame. Que constatez vous ?
 - Quel est le protocole MAC utilisé par les motes? Quel est son Channel check rate ?
 - Quelle est la fréquence de réveil de la couche radio d'un mote d'après la Timeline ?
 - Observer dans la fenêtre PowerTracker les valeurs de duty cycle de chaque nœud (Mote Radio duty Cycle).
 - Dans la fenêtre Timeline, identifier la valeur de la période de réveil d'un nœud.
 - Quel est le débit en kbits/s du protocole 802.15.4 ?
- Question 3 - Ouvrir maintenant la simulation : ~/contiki2.7/examples/ipv6/simple-udp-rpl/unicast-example.csc. Comparer le mode de transmission unicast au broadcast de la question précédente.

Exercice 2 : 6LoWPAN

Dans cette partie, nous nous intéressons au protocole 6LoWPAN qui est un format de compression des entêtes IPv6 et UDP sur 802.15.4 et décrit dans le RFC <https://tools.ietf.org/html/rfc6282>

- **Question 1** – Ouvrir la simulation `~/contiki2.7/examples/ipv6/rpl-udp/rpl-udp-powertrace.csc`
 - Ouvrir dans un éditeur de texte les deux fichiers `udp-client.c` et `udp-server.c` qui se trouvent dans le même répertoire
 - Quels sont les numéros des ports UDP utilisés par le client et le serveur?
 - Dans quelle variable du programme `udp-client.c` est stockée la réponse du serveur ?
 - Expliquer brièvement le fonctionnement de chacun des programmes.
- **Question 2** – Ouvrir les fenêtres Timeline, Mote output, Radio messages et lancer la simulation
 - Quelle est la durée de transmission d'un message entre le client et le serveur, entre le serveur et le client ?
 - Dans la fenêtre Radio messages, choisissez 6LoWAN Analyzer dans le menu Analyser.
 - Quels sont les différents messages échangés entre les nœuds ?
 - Quels sont les adresses IPv6 utilisées par les nœuds pour communiquer entre eux ?
 - Quel est le lien entre ces adresses IPv6 et celles présentes dans l'en-tête 802.15.4 ?
 - Expliquez les différents champs de l'en-tête IPHC utilisés par le protocole 6LoWPAN ? Il faut se référer au RFC 6282.

Exercice 3 : routage RPL

Dans cette partie nous nous intéressons au protocole RPL construisant les routes dans un WSN 6lowpan. RPL est décrit dans le RFC <https://tools.ietf.org/html/rfc6550>

- **Question 1** - Ouvrir la simulation `~/contiki2.7/examples/ipv6/rpl-collect/collect-tree-dense-noloss.csc`
 - Ouvrir dans un éditeur de texte les deux fichiers `udp-sink.c` et `udp-sender.c` se trouvant dans le même répertoire
 - Quelle est la valeur du duty cycle du nœud 1 ? Que signifie-t-elle ?
- **Question 2** - Observez sur la fenêtre Network la construction de l'arbre du routage RPL.
 - Quel est le nœud sink ?
 - Que représentent les flèches noires et les flèches bleues ?
 - A quel instant le nœud sink reçoit le premier message ? De quel nœud ?
 - Utiliser les fenêtres Mote output et Radio messages pour identifier l'arrivée d'un message sur le sink et son paquet 6LoWPAN correspondant. Examinez le contenu de ce paquet.
- **Question 3** - Nous nous intéressons maintenant aux messages utilisés par le protocole RPL pour construire un arbre de routage.
 - Identifier les différents messages utilisés par ce protocole.
 - Quelle est la valeur du rang (rank) annoncée par le nœud sink ?
 - Quelle est la valeur du rang du nœud 25 ?
 - De combien s'incrémente le rang entre un nœud et son parent préféré ?
 - Faites une capture d'écran de l'arbre de routage construit.
- **Question 4** - Déplacer maintenant le nœud 5 pour qu'il s'éloigne du nœud 2 et qu'il devienne plus proche au nœud 14.
 - Relancez la simulation. Choisissez le nœud 1, appuyer sur le bouton gauche et choisissez click button on sky 1.
 - Qu'observe-t-on sur l'arbre du routage RPL ?
 - Qu'observe-t-on dans la fenêtre Radio messages ?