

TP - OpenWSN - Simulation

Cours Protocoles sans-fil pour l'IoT critique.

3SN Parcours SEmblIoT

19 Nov. 2021

Katia Jaffrès Runser.

Dept SN, ENSEEIHT.

1- Déroulement du TP.

Voici les principales étapes du TP que vous pouvez réaliser. Il sera réalisé avec l'outil OpenWSN présenté dans la partie 2 de ce document.

Cet outil permet de simuler, d'émuler et de contrôler des réseaux d'objets implantant la pile protocolaire 6TiSCH vue en cours.

a. Connecter votre poste à Internet s'il ne l'est pas.

Pour cela, il faut :

- brancher sur une des prises Internet de la salle
- ouvrir un navigateur pour s'identifier sur le portail captif de l'N7 (prendre une adresse <http://www.enseeiht.fr>, puis se connecter avec votre compte sur le portail après avoir accepté les exceptions de sécurité)

Vérifier que vous avez accès au web : ouvrir www.openwsn.org

Vous pouvez aussi chercher ce sujet sur Moodle.

b. Lancer une simulation

Lancer une simulation qui comporte **5 noeuds avec l'outil OpenWSN présenté dans la partie.**

Consulter l'interface web du simulateur pour comprendre les différentes fenêtres.

Il existe des explications de manipulation de l'IHM dans ce tutoriel

<https://openwsn.atlassian.net/wiki/spaces/OW/pages/29196302/Kickstart+Linux>

Attention, il faut d'abord compiler le code du firmware.

- Représentation du réseau : modifier la topologie

Dans le menu Topology, le réseau de 5 noeuds est représenté.

On peut supprimer ou modifier le PER (packet error rate) d'un lien en cliquant dessus (click gauche).

Pour ajouter un lien entre deux noeuds, il faut les sélectionner successivement avec un click droit (attention, il faut sélectionner le ballon rouge, pas le carré bleu).

c. Lancer une premier routage RPL avec un seul DODAGroot.

Il faut sélectionner un noeud DODAGroot dans le menu Motes. Sélectionner

l'indice d'un des noeuds et cliquer sur 'Toggle DoDAGroot state'.

#Observations :

- Dans le terminal qui a lancé la simulation, on observe des logs. Que représentent les informations listées ici ?

Dans l'interface web, dans l'onglet Motes, on observe différentes informations relatives aux noeuds et aux échanges.

- Quel est le voisinage des noeuds (leur identifiant, leur rang RPL, etc.)
- Quels slots sont utilisés dans le simulateur pour les émissions et les réceptions par le DoDAGroot? Quel type de slots voit-on (partagé ou non) ?
- Modifier le PER d'un lien et voir l'impact de ce changement sur la connectivité et le routage.

d. Observer du traffic

#Utiliser Wireshark pour voir comment se comporte le réseau.

Dans OpenWSN, dans le menu Event Bus, il faut cocher la case 'Wireshark debug'

Puis, il faut lancer wireshark en root (sudo wireshark) et choisir l'interface **tun** pour observer ce qu'il se passe sur le réseau 6TiSCH.

Pour observer uniquement les trames du réseau, on peut configurer le filtrage de l'interface **tun** avec le filtre **zep** (ZigBee encapsulation protocol)

- Observer les échanges de trames avec Wireshark. Quels sont les types de trames observés ?
- Pour les beacons de la couche TSCH, observer le champ qui définit la slotframe : combien de liens existe-t-il ? Sont-ils en accès partagé ou dédié ?
- Sur quel slot/canal sont émis les enhanced beacons ?
- Sur quel slot/canal sont émis les packets de contrôle RPL ?
- Qui émet des données ? Sur quel canal ?
- Observer le DODAG obtenu avec RPL avec Wireshark.
 - Quel est le rang des noeuds ?
 - Quelles sont les métriques utilisées
- Observer le saut de fréquences (l'alternance des canaux utilisés sur le header 802.15.4.

e. Interroger les noeuds en IPv6

#Faire un ping6 sur les noeuds de la simulation depuis un nouveau terminal.

Attention, il faut changer l'application utilisée pour que le réseau ait une route pour répondre à la requête.

- Observer dans le terminal d'OpenWSN les logs du type **INFO 3 [ICMPv6ECHO] received an echo request**
- Dans le terminal du ping, vous observez des entrées telles que **10 bytes from bbbb::1415:92cc:0:3, icmp_seq=2 hlim=64(DUP!)**. Que signifie 'DUP' ? Est-ce normal ?
- Observer les statistiques sur les trames émises dans OpenWSN

(numéro de slots, taux de succès, etc.). Sur quel slot ces trames sont-elles envoyées ?

- Observer les trames du ping avec Wireshark.

f. Saut de fréquences.

Observer le saut de fréquences dans les trames Wireshark : il y a une différence entre le numéro de canal (virtuel) et les canaux réellement utilisés et observés dans les trames 802.15.4.

Trouver dans l'implantation de IEEE802.15.4e la définition de la séquence de canaux utilisés par le saut de fréquence dans cette implantation.

g. Comprendre la mise en place du schedule

Les fonctions et procédures permettant à la couche TSCH de mettre en place un schedule sont définies dans schedule.c.

La négociation du schedule est elle réalisée par le module sixtop. Observer les échanges 6-Top et lister les différentes cellules proposées dans l'échange et la cellule choisie par le parent.

2- L'outil OpenWSN.

Cet outil permet de simuler, d'émuler et de contrôler des réseaux d'objets implantant la pile protocolaire 6TiSCH vue en cours.

Cette partie liste les informations utiles pour manipuler OpenWSN.

#Site web du projet

<https://openwsn.atlassian.net/wiki/spaces/OW/pages>

#Pour l'installer chez soi (Linux, Windows, MacOS)

<https://openwsn.atlassian.net/wiki/spaces/OW/pages/12058660/Get+Started>

#Code @ Github

<https://github.com/openwsn-berkeley/>

#Répertoire d'installation d'OpenWSN sur les machines en C308 et C309

###

Tout l'environnement OpenWSN est installé dans le répertoire
/opt/openwsn

Il contient principalement trois sous-répertoires :

/opt/openwsn/openwsn-fw

/opt/openwsn/openvisualizer

/opt/openwsn/coap

Le répertoire FW contient les sources et le firmware qui tourne sur un noeud avec la pile protocolaire définie en langage C.

Le répertoire openvisualizer contient, entre autres, le simulateur qui utilise une version émulée du firmware définit dans le répertoire FW.

Le répertoire COAP contient le protocole de couche applicative utilisable pour consulter les données produites par les capteurs via une requête (protocole

similaire au protocole HTTP).

#Compiler the code du firmware et le lier au simulateur :

```
cd /opt/openwsn/openwsn-fw  
scons board=python toolchain=gcc oos_openwsn
```

Note : il faut re-compiler si on change le code d'un des protocoles de la pile 6TiSCH.

#voir toutes les options du script de compilation

```
scons --help
```

#Documentation Doxygen du firmware

Elle se trouve dans le répertoire openwsn-fw/build/docs/html.

Ouvrir **index.html**

Si elle n'est pas présente, il faut la générer avec

```
cd /opt/openwsn/openwsn-fw  
scons docs
```

#Démarrer le simulateur OpenVisualizer

#Pour démarrer le simulateur, il faut se placer dans le répertoire suivant

```
cd /opt/openwsn/openvisualizer
```

#Lancer la simulation

```
sudo scons runweb --sim
```

#Arrêter la simulation

```
quit
```

Si cette commande ne fonctionne pas, on tue le processus avec la commande suivante:

```
ps -ef | grep openVisualizer  
sudo kill -9 PID
```

#Pour voir les différentes options du lancement du simulateur

```
scons --help
```

Une fois la simulation lancée, on ne peut plus rien changer à la topologie.

#Par exemple :

--simCount=n Lance le simulateur sur un exemple qui comporte 'n' noeuds.

--simTopology=<linear|fully-meshed> Force la topologie initiale à *linear* ou *fully-meshed*. Il est possible de la modifier dans le simulateur.

#Pour accéder à l'interface graphique du simulateur,

Ouvrir la page web 127.0.0.1:8080 sur un navigateur.

#Ping d'un noeud qui tourne dans le simulateur

A partir d'un terminal, lancer la commande pour faire un ping sur l'adresse IPv6 d'un noeud simulé

ping6 -s 2 bbbb::1415:92cc:0:3

ping avec une payload de 2 octets.

Le réseau qui est généré a un préfixe bbbb:0000:0000:0000 par défaut. Le reste de l'adresse est donné par l'adresse MAC du capteur.

Attention : pour que ce ping6 fonctionne, il faut utiliser un application qui permet de router la requête vers une feuille. Pour cela, il faut observer des messages DAO (Destination Advertisement Object) qui permettent aux noeuds de créer des tables de routage.

#Modifier des paramètres du firmware (et donc d'un des protocoles de 6TiSCH)

Les fichiers C se trouvent dans le sous-répertoire suivant :

/opt/openwsn/openwsn-fw/openstack

L'ensemble des modules qui composent l'empilement protocolaire est décrit dans le module **openstak** (fichiers openstack.h et .c).

ATTENTION : Pour que les modifications soient prises en compte dans le simulateur, il faut RECOMPILER le firmware et relancer la simulation.

#Pour changer d'application

Le simulateur prévoit différentes applications. Leur code se trouve dans le répertoire

/opt/openwsn/openwsn-fw/openapps

Les applications suivantes permettent de :

uecho : répondre au ping. La couche transport est alors UDP.

cinfo : permet aux noeuds de répondre à une requête COAP (au dessus d'UDP) qui les interroge sur leur identité (nom de la carte)

#Pour jouer avec le schedule TSCH

On peut modifier les fichiers schedule.h et schedule.c (dans le répertoire **02b-MACHigh**).

- Certains paramètres sont modifiables dans schedule.h. Par exemple (donnés en nombre de slots) :

#define SLOTFRAME_LENGTH 101 //la taille de la slotframe

#define SCHEDULE_MINIMAL_6TISCH_ACTIVE_CELLS 1

Cette dernière constante donne le nombre de slots partagés créés à l'initialisation du réseau pour transmettre les EB et les trames de contrôle RPL. Plus ce nombre est élevé, moins les collisions sur ce slot partagé sont importantes.

#define NUMSLOTSOFF 10

Cette constante donne le nombre maximal de slots dédiés qu'un noeud peut allouer dans la slotframe, en plus des slots partagés (6TISCH_ACTIVE_CELLS)

- Le schedule est définit dans schedule.c

Il est enregistré dans le tableau la variable schedule_vars de type schedule_vars_t (type enregistrement définit dans schedule.h)

schedule_vars_t schedule_vars

Il n'est réellement initialisé qu'une fois le DAG root choisit avec la procédure

`schedule_startDAGroot().`