TP 2 – Time Slotted Channel Hopping (TSCH)

Dans ce TP vous allez utiliser l'Equipex FIT-IoT Lab pour analyser les performances du protocole TSCH (*Time Slotted Channel hopping*) de IEEE 802.15.4. Connectez-vous à la plateforme via l'interface web à l'aide des identifiants qui vous ont été communiqués :

https://www.iot-lab.info/testbed

et ajoutez votre clef publique SSH (dans `Account / SSH Keys`) afin de pouvoir vous connectez via SSH ultérieurement.

Exercice 1

Dans cet exercice, vous allez déployer un *firmware* pré-configuré sur deux nœuds de la plateforme et interagir avec ces nœuds. Tout se déroulera en ligne via votre navigateur web. Pour ce faire, vous pouvez suivre le tutoriel disponible ici :

https://www.iot-lab.info/legacy/tutorials/getting-started-tutorial/

Vous devez utiliser la plateforme de Strasbourg et limiter la durée de l'expérience à 10 minutes. Vous devez également sélectionner les nœuds assignés à votre binôme.

Exercice 2

Dans cet exercice vous allez compiler et déployer un *firmware* depuis l'interface en ligne de commande. Ce *firmware* sera issu de contiki-os, un système d'exploitation pour les objets contraints. Suivez les instructions suivantes :

1. Connectez vous au serveur via ssh :
   `my_computer$ ssh <login>@strasbourg.iot-lab.info`
2. Entrez la commande suivante pour enregistrer votre mot de passe :
   `iotlab-auth -u <login>`
3. Suivez le tutoriel disponible ici :
   https://www.iot-lab.info/legacy/tutorials/contiki-ng-compilation/
   Attention, entre les étapes 2 et 3 entrez les commandes :
   ```
   cd ~/iot-lab/parts/iot-lab-contiki-ng/contiki-ng
   git submodule update --init
   ```
   Pour soumettre votre expérience vous utiliserez l'interface en ligne de commande en prenant soin d'utiliser la plateforme de Strasbourg et l'un des nœuds qui ont été assignés à votre binôme.
4. Attendez que votre expérience soit lancée via la commande :
   `iotlab-experiment wait`
5. Affichez les ressources utilisées par votre expérience via la commande :
   `iotlab-experiment get -p`
6. Connectez-vous au port série du nœud via la commande :
   `nc m3-<node_id> 20000`
7. Vous pouvez créer un tunnel SSH entre votre poste et le serveur via :
   `my_computer$ ssh -L 20000:m3-<id>:20000 <login>@strasbourg.iot-lab.info`
   et dans un autre terminal lancez la commande :
   `nc localhost 20000`

Exercice 3

Dans cet exercice vous allez utiliser des *profiles* qui déterminent ce que fait le nœud de contrôle d'un nœud durant l'expérience.

Avant de vous lancer dans les tutoriels suivants, vous devez configurer votre poste pour pouvoir générer les graphiques à distance en activant le déport d'affichage avec SSH (*X11 forwarding*) :

- pour les utilisateurs Linux il n'y a rien de spécifique à faire
- pour les utilisateurs sous MAC OSX il faut installer `Quartz` :
  https://www.xquartz.org/
- pour les utilisateurs sous Windows, il faut installer `Xming` :
  https://sourceforge.net/projects/xming/
  et suivre le tutoriel pour la configuration de `Putty` :
  https://quick-tutoriel.com/comment-utiliser-loption-x11-forwarding-sous-putty/

Dans la suite, vous utiliserez exclusivement le site de Strasbourg et les nœuds qui vous ont été assignés.

1. Ce premier tutoriel vous montre comment surveiller la consommation d'énergie :
   https://www.iot-lab.info/legacy/tutorials/monitoring-consumption-m3/
   Note : dans l'étape 10, on vous indique le chemin :
   ~/.iot-lab/<experimentid>/consumption/m3-<id>.oml
   qu'il faut remplacer par :
   ~/.iot-lab/<experimentid>/consumption/m3_<id>.oml
2. Ce second tutoriel vous montre comment surveiller l'activité radio. Vous pouvez directement aller à l'étape 4 sans recompiler le firmware. Vous pouvez tous utiliser le canal 11 afin de voir le trafic de vos collègues.
   https://www.iot-lab.info/legacy/tutorials/monitoring-radio-m3/
3. Enfin ce troisième tutoriel vous montre comment générer des traces wireshark :
   https://www.iot-lab.info/legacy/tutorials/monitoring-sniffer-m3/

Exercice 4

Dans cet exercice, vous allez réaliser une expérience simple pour vous familiariser avec le protocole TSCH et l'ordonnancement Orchestra.

1. Créez un programme `sender` qui envoie périodiquement un datagramme UDP au coordinateur de PAN. Ce programme doit utiliser IPv6, le protocole de routage RPL et TSCH au niveau MAC. L'ordonnancement TSCH doit être réalisé avec Orchestra. Vous pouvez utiliser la configuration par défaut d'Orchestra.
2. Créez un programme `coordinateur` qui joue le rôle de coordinateur de PAN 802.15.4. Pour chaque datagramme UDP reçu, il répond par un nouveau datagramme UDP à l'expéditeur. Ce programme utilise IPv6, le protocole de routage RPL (il fera office de racine de l'arbre de routage) et TSCH au niveau MAC. L'ordonnancement TSCH doit être réalisé avec Orchestra. Vous pouvez utiliser la configuration par défaut d'Orchestra.
3. Modifiez le PAN ID de votre réseau dans le fichier `project-conf.h` pour utiliser l'identifiant du nœud le plus grand qui vous a été assigné (ex : m3-128 => PAN ID : 0x128).
4. Lancez une expérimentation avec 2 nœuds (un coordinateur et un nœud émetteur). Analysez les logs produits par chacun des nœuds. Détaillez l'ordonnancement réalisé par Orchestra (nombre de slotframes, priorité, taille, offset canal, règle, etc.). Les logs générés par TSCH sont présentés en annexe.

Exercice 5

Vous devez maintenant réaliser une analyse de performance du protocole TSCH et de l'ordonnancement Orchestra. C'est à vous de définir les scénarios (modèles de trafic, comparaison avec un autre protocole MAC, etc.), les métriques que vous souhaitez évaluer (PDR, délai, etc.), la configuration de l'ordonnancement (règles,

nombre de slotframes, taille, etc.), la collection des résultats et leur mise en forme. Vous attacherez la plus grande importance à la qualité de vos mesures : valeurs significatives, élimination des variations et des perturbations, etc.

Vous devrez utiliser un dépôt GIT sur git.unistra.fr sur lequel vous devrez déposer :

- les programmes utilisés pour réaliser les tests (fichiers C, Makefile, project-conf.h) ;
- les scripts et/ou programmes utilisés pour automatiser le lancement de vos expériences sur la plateforme FIT IOT-LAB ;
- les fichiers résultats bruts (avant tout traitement) ;
- les scripts et/ou programmes utilisés pour générer les figures de résultat à partir des résultats bruts.

Vous devrez donner un accès en lecture à votre enseignant. La régularité et la quantité de travail seront mesurées par rapport aux activités sur le dépôt et prises en compte dans la notation.

Vous présenterez votre travail dans un rapport au format PDF à déposer sur Moodle. Les sources de ce document devront également être disponibles sur votre dépôt GIT. Ce document devra contenir :

- une description de vos scénarios et leur justification ;
- la définition des métriques retenues, leur pertinence dans le contexte de cette étude, et comment vous les avez mesurées ;
- les résultats obtenus (sous la forme de graphiques) et la méthodologie pour leur génération ;
- une analyse détaillée des résultats obtenus.

Le rapport est limité à 10 pages annexes comprises.

Note : la configuration utilisée par défaut dans Orchestra se trouve dans le fichier :

`os/services/orchestra/orchestra-conf.h`

Vous pouvez définir vos propres macros `ORCHESTRA_CONF_*` dans le fichier `project-conf.h` pour modifier cette configuration.

Vous trouverez également plus d'informations sur l'implémentation de TSCH ici :

https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-TSCH-and-6TiSCH

Annexes

Connexion au port série d'un nœud :

```
nc m3-<id> 20000
```

Agrégation de tous les ports série des nœuds de l'expérience courante dans un même terminal :

```
serial_aggregator
```

Soumission d'une expérience :

```
iotlab-experiment submit -n <EXPERIMENT NAME> -d <DURATION IN MINS> -l
<SITE NAME, ARCHI, NODES ID LIST, FIRMWARE FILE NAME, PROFILE FILE NAME>
```

Génération de traces wireshark :

```
sniffer_aggregator -l <SITE>,m3,<ID> -o <OUTPUT PCAP FILE>
```

Format de sortie des logs de TSCH :

```
                     burst count     canal   0 == Eb, 1 == Data                      status
                          ^            ^     ^         src                             ^
                          |            |     |          ^                              |
{asn 00.1193 link  0 397   0 132  0 ch 20} bc-0-0 tx LL-a484->LL-NULL, len 35, seq 195, st 0 1
        |          |  |           |        v    | |              |                          | |
        v          |  |           |   broadcast | v              v                          | v
 Absolute Slot     |  |           v             v transmission   dest                       | tx
      Number       |  v      offset canal       security level                           v attempt
                   v  taille de la slotframe                                  0 == SUCCESS
        n° de slotframe (+ petit + prioritaire)                               2 == NOACK
```